

计算机系统结构课程实验

总结报告

实验题目：动、静态流水线设计与性能对比分析

一、实验环境部署与硬件配置说明

1、实验环境

- 操作系统: Windows10
- 编程语言: Verilog HDL
- 开发环境: Vivado 2016.2、Modelsim

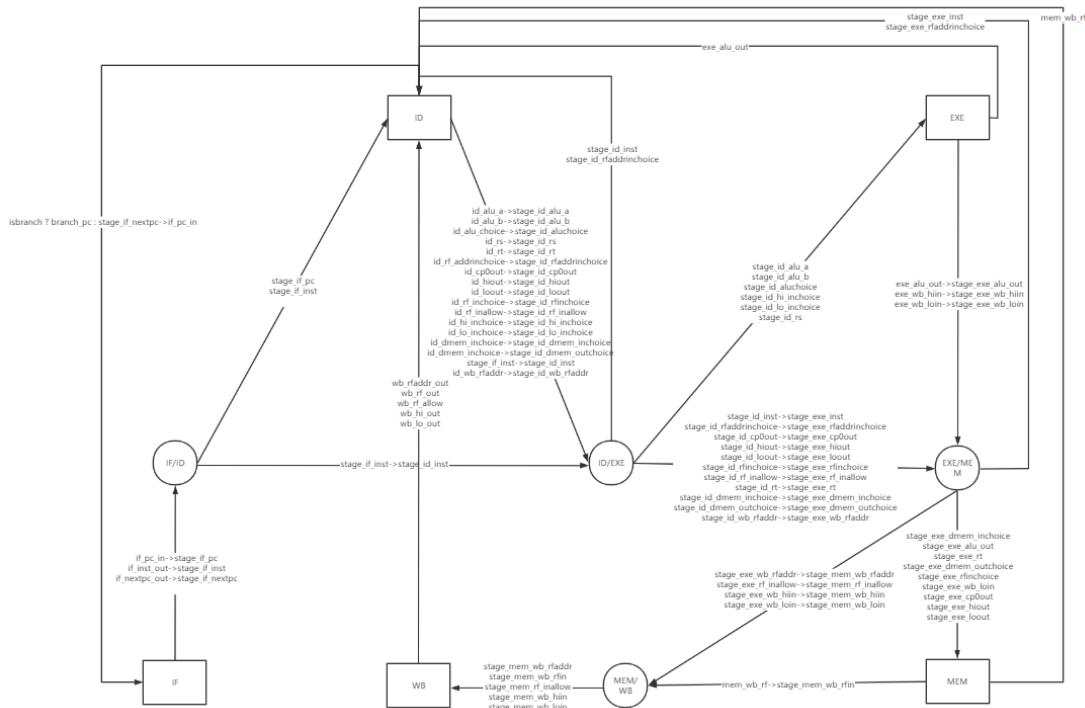
2、硬件配置

- 处理器: i5-8300H
- 内存: 8GB
- 测试板: Xilinx NEXYS4 DDR

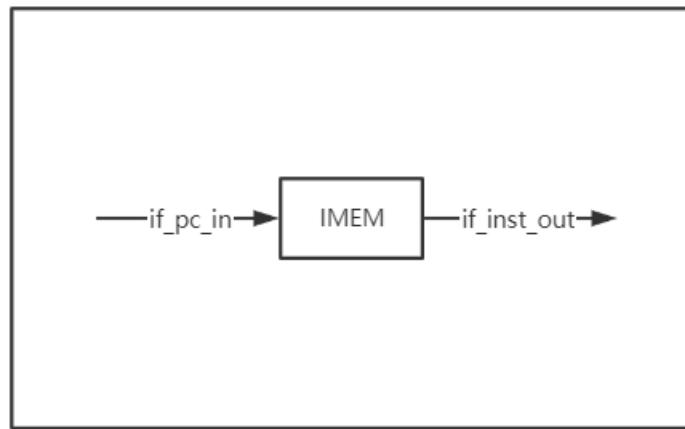
二、实验的总体结构

1、动态流水线的总体结构

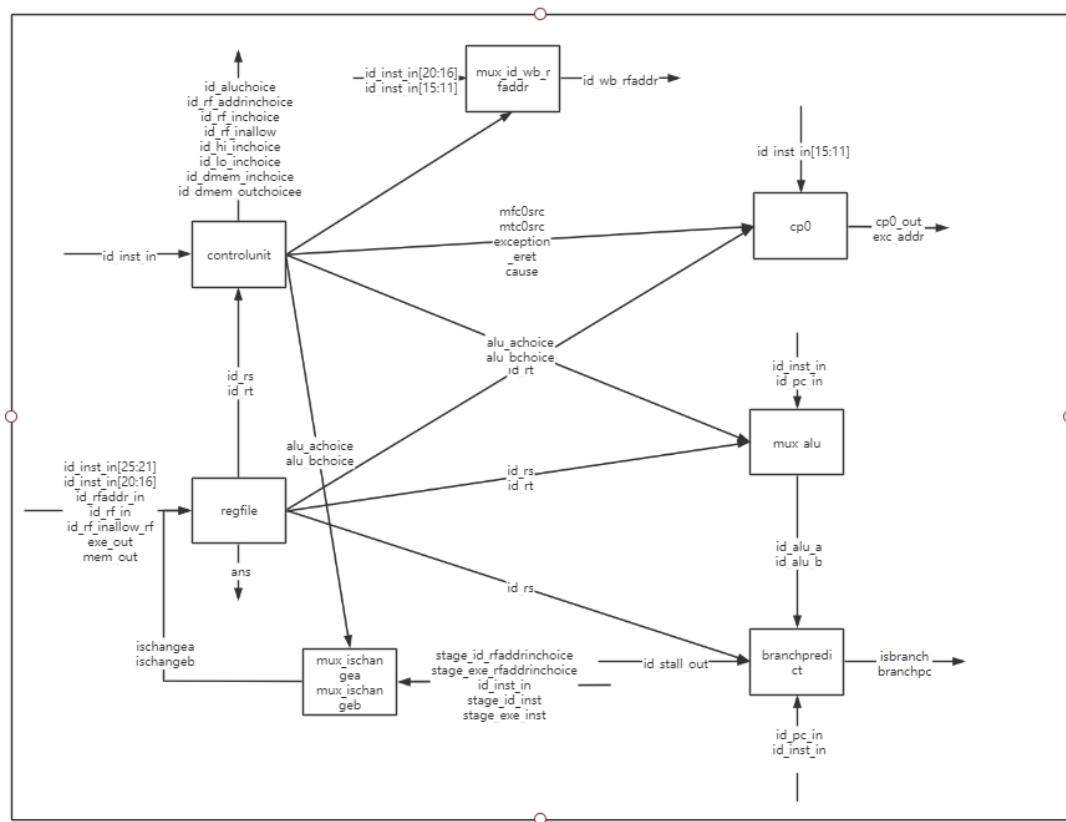
1.1 总体架构



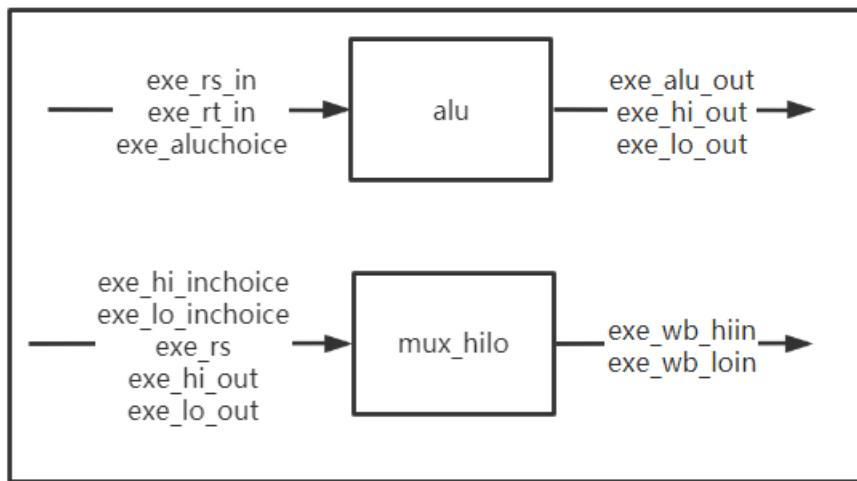
1.2 IF 结构



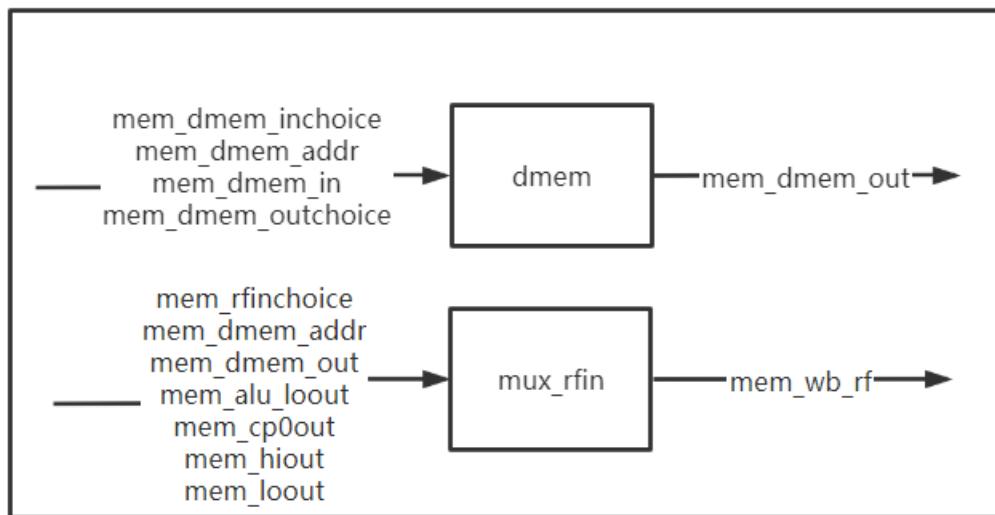
1.3 ID 结构



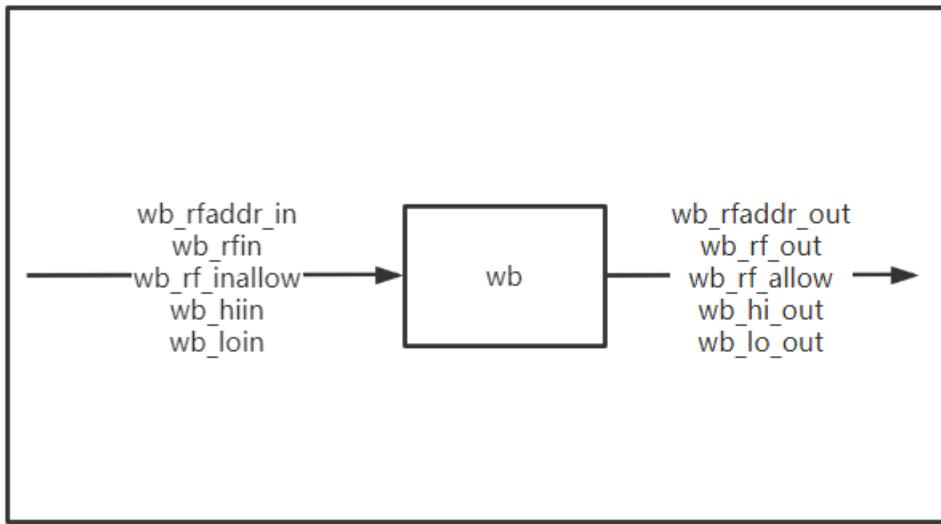
1.4EXE 结构



1.5MEM 结构

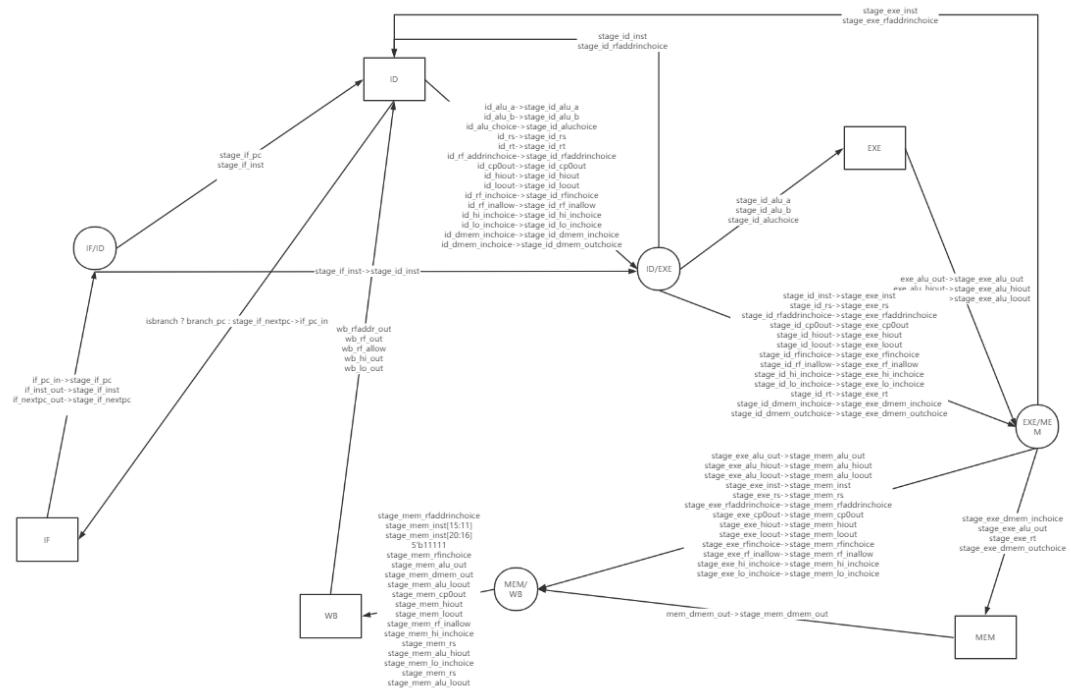


1.6WB 结构

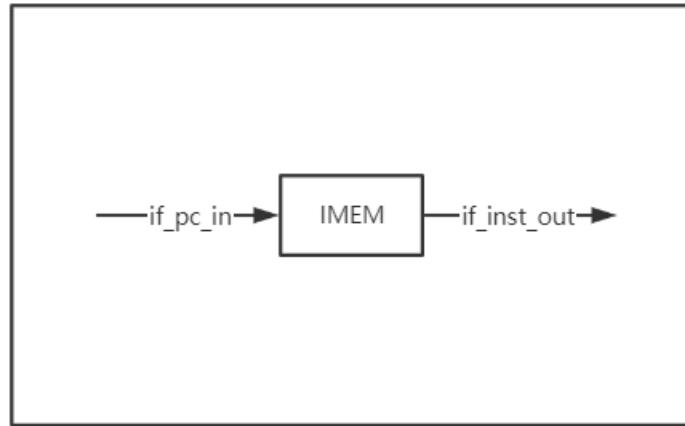


2、静态流水线的总体结构

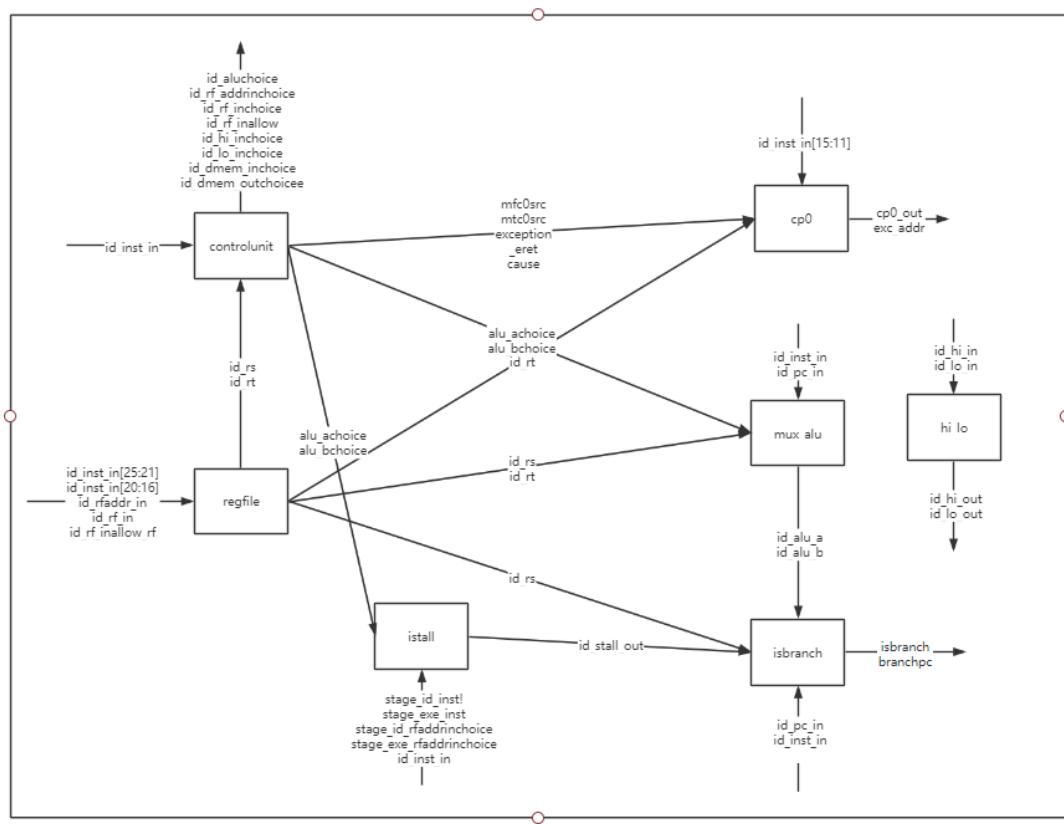
2.1 总体架构



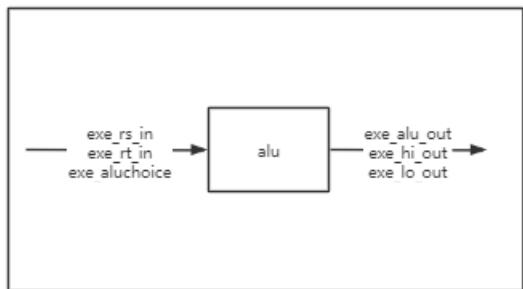
2.2IF 结构



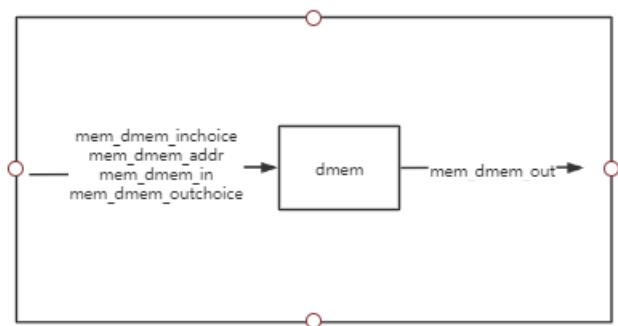
2.3ID 结构



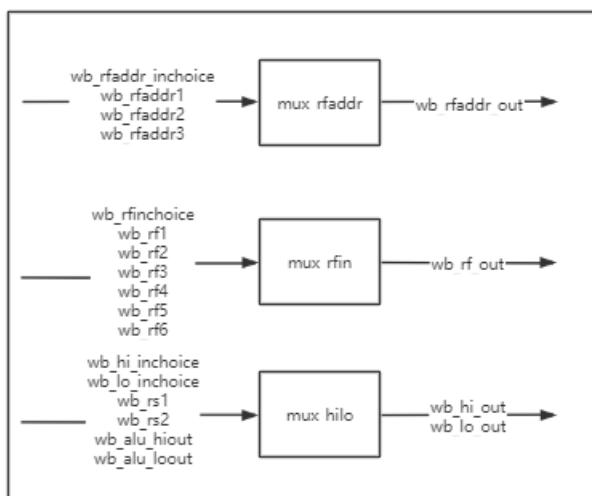
2.4EXE 结构



2.5MEM 结构



2.6WB 结构



三、总体架构部件的解释说明

1、动态流水线总体结构部件的解释说明

动态流水线 CPU 分为 9 个大模块，分别是 IF、IF/ID、ID、ID/EXE、EXE、EXE/MEM、MEM、MEM/WB、WB，下面分别解释说明这 9 个模块；

1.1IF 模块

部件：指令存储器 imem；

输入：当前 pc

输出：当前 pc 对应的指令和 pc+4；

接口定义：

```
module Instfetch(
    input [31:0] if_pc_in,
    output [31:0] if_inst_out,
    output [31:0] if_nextpc_out
);
```

1.2IF/ID 模块

作为 IF/ID 级间的流水寄存器；

Verilog 代码描述：

```
always @ (negedge clk_in)
begin
    if(reset)
        begin
            stage_if_inst <= 0;
            stage_if_pc <= 0;
            stage_if_nextpc <= 0;
        end
    else if (!id_stall_out)
        begin
            stage_if_inst <= if_inst_out;//当前指令 inst
            stage_if_pc <= if_pc_in;//当前指令 pc，用于 ID
            stage_if_nextpc <= if_nextpc_out;//下一指令 pc，如果没有跳转，则赋给 if_pc_in
        end
end
```

1.3 ID 模块

部件：控制单元 controlunit、寄存器堆 regfile、cp0、寄存器 hi_lo、运算器输入的多路选择器 mux_alu、判断是否暂停 isstall、判断是否分支跳转 isbranch；

输入：WB 级传回的写信号、写地址、写数据，IF 级传递的值、ID/EXE 和 EXE/MEM 传回的用于判断是否暂停的值，EXE 级和 MEM 级产生的结果前推；

输出：各类控制信号，向 EXE 级传递的值；

```
module Instdecode(
    input clk,
    input reset,
    input [31:0]id_pc_in,
    input [31:0]id_inst_in,
    input [4:0]id_rfaddr_in,
    input [31:0]id_rf_in,
    input id_rf_inallow_rf,
    input [31:0]id_hi_in,
    input [31:0]id_lo_in,

    input [31:0]stage_id_inst,// 
    input [1:0]stage_id_rfaddrinchoice,
    input [31:0]stage_exe_inst,
    input [1:0]stage_exe_rfaddrinchoice,
    //传入 exe_out、mem_out
    input [31:0]exe_out,
    input [31:0]mem_out,
    output [31:0]id_alu_a,
    output [31:0]id_alu_b,
    output [31:0]id_rs,
    output [31:0]id_rt,
    output [4:0]id_aluchoice,
    output [1:0]id_rf_addrinchoice,
    output [2:0]id_rf_inchoice,
    output id_rf_inallow,
    output id_hi_inchoice,
    output id_lo_inchoice,
    output isbranch,
    output [31:0]branch_pc,
    output id_stall_out,
    output [31:0]cp0_out,
    output [31:0]id_hi_out,
```

```

output [31:0]id_lo_out, //来自 writeback

output [1:0]id_dmem_inchoice,
output [2:0]id_dmem_outchoice,
output [31:0]ans,

output [4:0]id_wb_rfaddr//wb
);

```

1.4ID/EXE 模块

作为 ID/EXE 级间的流水寄存器;

Verilog 代码描述:

```

always @ (negedge clk_in)
begin
    if(reset)
        begin
            stage_id_alu_a <= 0;
            stage_id_alu_b <= 0;
            stage_id_aluchoice <= 0;
            stage_id_inst <= 0;
            stage_id_rs <= 0;
            stage_id_rt <= 0;
            stage_id_rfaddrinchoice <= 0;
            stage_id_cp0out <= 0;
            stage_id_hiout <= 0;
            stage_id_loout <= 0;
            stage_id_rfchoice <= 0;
            stage_id_rf_inallow <= 0;
            stage_id_hi_inchoice <= 0;
            stage_id_lo_inchoice <= 0;
            stage_id_dmem_inchoice <= 0;
            stage_id_dmem_outchoice <= 0;

            //writeback 用
            stage_id_wb_rfaddr <= 0;
        end
    else
        begin
            if (!id_stall_out)
                begin
                    stage_id_alu_a <= id_alu_a;
                    stage_id_alu_b <= id_alu_b;

```

```

stage_id_aluchoice <= id_alu_choice;

//writeback 用
stage_id_inst <= stage_if_inst;
stage_id_rs <= id_rs;
stage_id_rt <= id_rt;
stage_id_rfaddrinchoice <= id_rf_addrinchoice;
stage_id_cp0out <= id_cp0out;
stage_id_hiout <= id_hiout;
stage_id_loout <= id_loout;
stage_id_rfchoice <= id_rf_inchoice;
stage_id_rf_inallow <= id_rf_inallow;
stage_id_hi_inchoice <= id_hi_inchoice;
stage_id_lo_inchoice <= id_lo_inchoice;
stage_id_dmem_inchoice <= id_dmem_inchoice;
stage_id_dmem_outchoice <= id_dmem_inchoice;

//writeback 用
stage_id_wb_rfaddr <= id_wb_rfaddr;
end
end
end

```

1.5EXE 模块

部件：运算器 alu(内部封装了乘法器、除法器、前导零计算等);

输入：ID 级传递的控制信号和各类源操作数的值;

输出：计算结果和 hi_lo 要写入的结果;

接口定义：

```

module Execute(
    input [31:0]exe_rs_in,
    input [31:0]exe_rt_in,
    input [4:0]exe_aluchoice,

```

```

//writeback
input exe_hi_inchoice,
input exe_lo_inchoice,
input [31:0]exe_rs,
```

```
output [31:0]exe_alu_out,
```

```

//writeback
output [31:0]exe_wb_hiin,
```

```
    output [31:0]exe_wb_loin  
);
```

1.6EXE/MEM 模块

作为 EXE/MEM 级间的流水寄存器;

Verilog 代码描述:

```
always @ (negedge clk_in)  
begin  
    if(reset)  
        begin  
            stage_exe_alu_out <= 0;  
            stage_exe_inst <= 0;  
            stage_exe_rfaddrinchoice <= 0;  
            stage_exe_cp0out <= 0;  
            stage_exe_hiout <= 0;  
            stage_exe_loout <= 0;  
            stage_exe_rfinchoice <= 0;  
            stage_exe_rf_inallow <= 0;  
  
            stage_exe_rt <= 0;  
            stage_exe_dmem_inchoice <= 0;  
            stage_exe_dmem_outchoice <= 0;  
  
            //writaback 用  
            stage_exe_wb_rfaddr <= 0;  
            stage_exe_wb_hiin <= 0;  
            stage_exe_wb_loin <= 0;  
        end  
    else  
        begin  
            stage_exe_alu_out <= exe_alu_out;  
  
            stage_exe_inst <= stage_id_inst;  
            stage_exe_rfaddrinchoice <= stage_id_rfaddrinchoice;  
            stage_exe_cp0out <= stage_id_cp0out;  
            stage_exe_hiout <= stage_id_hiout;  
            stage_exe_loout <= stage_id_loout;  
            stage_exe_rfinchoice <= stage_id_rfinchoice;  
            stage_exe_rf_inallow <= stage_id_rf_inallow;  
  
            stage_exe_rt <= stage_id_rt;  
            stage_exe_dmem_inchoice <= stage_id_dmem_inchoice;
```

```

stage_exe_dmem_outchoice <= stage_id_dmem_outchoice;

//writaback 用
stage_exe_wb_rfaddr <= stage_id_wb_rfaddr;
stage_exe_wb_hiin <= exe_wb_hiin;
stage_exe_wb_loin <= exe_wb_loin;
end
end

```

1.7MEM 模块

部件：数据存储器 dmem；

输入：当前 EXE 级计算结果和传递的控制信号；

输出：读出的结果和 regfile 要写入的数据；

接口定义：

```

module Memory(
    input clk,
    input [1:0] mem_dmem_inchoice,
    input [31:0] mem_dmem_addr,//输入输出用一个地址
    input [31:0] mem_dmem_in,
    input [2:0] mem_dmem_outchoice,
    input [2:0]mem_rfchoice,//writeback
    input [31:0]mem_alu_loout,
    input [31:0]mem_cp0out,
    input [31:0]mem_hiout,
    input [31:0]mem_loout,
    output [31:0] mem_wb_rf
);

```

1.8MEM/WB 模块

作为 MEM/WB 级间的流水寄存器；

Verilog 代码描述：

```

always @ (negedge clk_in)
begin
    if(reset)
        begin
            stage_mem_wb_rfaddr <= 0;
            stage_mem_wb_rfin <= 0;
            stage_mem_rf_inallow <= 0;
            stage_mem_wb_hiin <= 0;
            stage_mem_wb_loin <= 0;
        end
end

```

```

    else
begin
    stage_mem_wb_rfaddr <= stage_exe_wb_rfaddr;
    stage_mem_wb_rfin <= mem_wb_rf;
    stage_mem_rf_inallow <= stage_exe_rf_inallow;
    stage_mem_wb_hiin <= stage_exe_wb_hiin;
    stage_mem_wb_loin <= stage_exe_wb_loin;
end
end

```

1.9WB 模块

部件：各类多路选择器；

输入：各类计算结果；

输出：写回 ID 级的写信号、写地址、写数据；

接口定义：

```

module Writeback(
    input [4:0]wb_rfaddr_in,
    input [31:0]wb_rfin,
    input wb_rf_inallow,
    input [31:0]wb_hiin,
    input [31:0]wb_loin,
    output [4:0]wb_rfaddr_out,
    output [31:0]wb_rf_out,
    output wb_rf_allow,
    output [31:0]wb_hi_out,
    output [31:0]wb_lo_out
);

```

2、静态流水线总体结构部件的解释说明

静态流水线 CPU 分为 9 个大模块，分别是 IF、IF/ID、ID、ID/EXE、EXE、EXE/MEM、MEM、MEM/WB、WB，下面分别解释说明这 9 个模块；

2.1IF 模块

部件：指令存储器 imem；

输入：当前 pc

输出：当前 pc 对应的指令和 pc+4；

接口定义：

```

module InstFetch(
    input [31:0] if_pc_in,

```

```

    output [31:0] if_inst_out,
    output [31:0] if_nextpc_out
);

```

2.2IF/ID 模块

作为 IF/ID 级间的流水寄存器；

Verilog 代码描述：

```

always @ (negedge clk)
begin
    if(reset==1 || isbranch==1)
        begin
            stage_if_inst <= 0;
            stage_if_pc <= 0;
            stage_if_nextpc <= 0;
        end
    else if (!id_stall_out)
        begin
            stage_if_inst <= if_inst_out;//当前指令 inst
            stage_if_pc <= if_pc_in;//当前指令 pc, 用于 ID
            stage_if_nextpc <= if_nextpc_out;//下一指令 pc, 如果没有跳转, 则赋给 if_pc_in
        end
end

```

2.3ID 模块

部件：控制单元 controlunit、寄存器堆 regfile、cp0、寄存器 hi_lo、运算器输入的多路选择器 mux_alu、判断是否暂停 isstall、判断是否分支跳转 isbranch；

输入：WB 级传回的写信号、写地址、写数据，IF 级传递的值、ID/EXE 和 EXE/MEM 传回的用于判断是否暂停的值；

输出：各类控制信号，向 EXE 级传递的值；

接口定义：

```

module InstDecode(
    input clk,
    input reset,
    input [31:0]id_pc_in,
    input [31:0]id_inst_in,
    input [4:0]id_rfaddr_in,
    input [31:0]id_rf_in,
    input id_rf_inallow_rf,
    input [31:0]id_hi_in,
    input [31:0]id_lo_in,
    input [31:0]stage_id_inst,

```

```

input [1:0]stage_id_rfaddrinchoice,
input [31:0]stage_exe_inst,
input [1:0]stage_exe_rfaddrinchoice,
output [31:0]id_alu_a,
output [31:0]id_alu_b,
output [31:0]id_rs,
output [31:0]id_rt,
output [4:0]id_aluchoice,
output [1:0]id_rf_addrinchoice,
output [2:0]id_rf_inchoice,
output id_rf_inallow,
output id_hi_inchoice,
output id_lo_inchoice,
output isbranch,
output [31:0]branch_pc,
output id_stall_out,
output [31:0]cp0_out,
output [31:0]id_hi_out,
output [31:0]id_lo_out, //要用到 rf_in 里, 在 writeback
output [1:0]id_dmem_inchoice,
output [2:0]id_dmem_outchoice
);

```

2.4ID/EXE 模块

作为 ID/EXE 级间的流水寄存器;

Verilog 代码描述:

```

always @ (negedge clk)
begin
  if(reset==1)
    begin
      stage_id_alu_a <= 0;
      stage_id_alu_b <= 0;
      stage_id_aluchoice <= 0;
      stage_id_inst <= 0;
      stage_id_rs <= 0;
      stage_id_rt <= 0;
      stage_id_rfaddrinchoice <= 0;
      stage_id_cp0out <= 0;
      stage_id_hiout <= 0;
      stage_id_loout <= 0;
      stage_id_rfchoice <= 0;
      stage_id_rf_inallow <= 0;
    end
  end
end

```

```

    stage_id_hi_inchoice <= 0;
    stage_id_lo_inchoice <= 0;
    stage_id_dmem_inchoice <= 0;
    stage_id_dmem_outchoice <= 0;
end
else if (!id_stall_out)
begin
    stage_id_alu_a <= id_alu_a;
    stage_id_alu_b <= id_alu_b;
    stage_id_aluchoice <= id_alu_choice;
    stage_id_inst <= stage_if_inst;
    stage_id_rs <= id_rs;
    stage_id_rt <= id_rt;
    stage_id_rfaddrinchoice <= id_rf_addrinchoice;
    stage_id_cp0out <= id_cp0out;
    stage_id_hiout <= id_hiout;
    stage_id_loout <= id_loout;
    stage_id_rfchoice <= id_rf_inchoice;
    stage_id_rf_inallow <= id_rf_inallow;
    stage_id_hi_inchoice <= id_hi_inchoice;
    stage_id_lo_inchoice <= id_lo_inchoice;
    stage_id_dmem_inchoice <= id_dmem_inchoice;
    stage_id_dmem_outchoice <= id_dmem_inchoice;
end
end

```

2.5EXE 模块

部件：运算器 alu(内部封装了乘法器、除法器、前导零计算等);

输入：ID 级传递的控制信号和各类源操作数的值；

输出：计算结果；

接口定义：

```

module Execute(
    input [31:0]exe_rs_in,
    input [31:0]exe_rt_in,
    input [4:0]exe_aluchoice,
    output [31:0]exe_alu_out,
    output [31:0]exe_hi_out,
    output [31:0]exe_lo_out
);

```

2.6EXE/MEM 模块

作为 EXE/MEM 级间的流水寄存器；

Verilog 代码描述：

```
always @ (negedge clk)
begin
    if(reset==1)
        begin
            stage_exe_alu_out <= 0;
            stage_exe_alu_hiout <= 0;
            stage_exe_alu_loout <= 0;
            stage_exe_inst <= 0;
            stage_exe_rs <= 0;
            stage_exe_rfaddrinchoice <= 0;
            stage_exe_cp0out <= 0;
            stage_exe_hiout <= 0;
            stage_exe_loout <= 0;
            stage_exe_rfinchoice <= 0;
            stage_exe_rf_inallow <= 0;
            stage_exe_hi_inchoice <= 0;
            stage_exe_lo_inchoice <= 0;
            stage_exe_rt <= 0;
            stage_exe_dmem_inchoice <= 0;
            stage_exe_dmem_outchoice <= 0;
        end
    else
        begin
            stage_exe_alu_out <= exe_alu_out;
            stage_exe_alu_hiout <= exe_alu_hiout;
            stage_exe_alu_loout <= exe_alu_loout;
            stage_exe_inst <= stage_id_inst;
            stage_exe_rs <= stage_id_rs;
            stage_exe_rfaddrinchoice <= stage_id_rfaddrinchoice;
            stage_exe_cp0out <= stage_id_cp0out;
            stage_exe_hiout <= stage_id_hiout;
            stage_exe_loout <= stage_id_loout;
            stage_exe_rfinchoice <= stage_id_rfinchoice;
            stage_exe_rf_inallow <= stage_id_rf_inallow;
            stage_exe_hi_inchoice <= stage_id_hi_inchoice;
            stage_exe_lo_inchoice <= stage_id_lo_inchoice;
            stage_exe_rt <= stage_id_rt;
            stage_exe_dmem_inchoice <= stage_id_dmem_inchoice;
            stage_exe_dmem_outchoice <= stage_id_dmem_outchoice;
        end
end
```

```
    end  
end
```

2.7MEM 模块

部件：数据存储器 dmem；
输入：当前 EXE 级计算结果和传递的控制信号；
输出：读出的结果；
接口定义：

```
module Memory(  
    input clk,  
    input [1:0] mem_dmem_inchoice,  
    input [31:0] mem_dmem_addr,//输入输出用一个地址  
    input [31:0] mem_dmem_in,  
    input [2:0] mem_dmem_outchoice,  
    output [31:0] mem_dmem_out  
);
```

2.8MEM/WB 模块

作为 MEM/WB 级间的流水寄存器；

Verilog 代码描述：

```
always @ (negedge clk)  
begin  
    if(reset==1)  
        begin  
            stage_mem_dmem_out <= 0;  
            stage_mem_alu_out <= 0;  
            stage_mem_alu_hiout <= 0;  
            stage_mem_alu_loout <= 0;  
            stage_mem_inst <= 0;  
            stage_mem_rs <= 0;  
            stage_mem_rfaddrinchoice <= 0;  
            stage_mem_rfinchoice <= 0;  
            stage_mem_rf_inallow <= 0;  
            stage_mem_hi_inchoice <= 0;  
            stage_mem_lo_inchoice <= 0;  
        end  
    else  
        begin  
            stage_mem_dmem_out <= mem_dmem_out;  
            stage_mem_alu_out <= stage_exe_alu_out;  
            stage_mem_alu_hiout <= stage_exe_alu_hiout;  
        end  
end
```

```

    stage_mem_alu_loout <= stage_exe_alu_loout;
    stage_mem_inst <= stage_exe_inst;
    stage_mem_rs <= stage_exe_rs;
    stage_mem_rfaddrinchoice <= stage_exe_rfaddrinchoice;
    stage_mem_cp0out <= stage_exe_cp0out;
    stage_mem_hiout <= stage_exe_hiout;
    stage_mem_loout <= stage_exe_loout;
    stage_mem_rfinchoice <= stage_exe_rfinchoice;
    stage_mem_rf_inallow <= stage_exe_rf_inallow;
    stage_mem_hi_inchoice <= stage_exe_hi_inchoice;
    stage_mem_lo_inchoice <= stage_exe_lo_inchoice;
end
end

```

2.9WB 模块

部件：各类多路选择器；

输入：各类计算结果和控制信号；

输出：写回 ID 级的写信号、写地址、写数据；

接口定义：

```

module Writeback(
    input [1:0]wb_rfaddr_inchoice,
    input [4:0]wb_rfaddr1,
    input [4:0]wb_rfaddr2,
    input [4:0]wb_rfaddr3,
    input [2:0]wb_rfinchoice,
    input [31:0]wb_rf1,
    input [31:0]wb_rf2,
    input [31:0]wb_rf3,
    input [31:0]wb_rf4,
    input [31:0]wb_rf5,
    input [31:0]wb_rf6,
    input wb_rf_inallow,
    input wb_hi_inchoice,
    input [31:0]wb_rs1,
    input [31:0]wb_alu_hiout,
    input wb_lo_inchoice,
    input [31:0]wb_rs2,
    input [31:0]wb_alu_loout,
    output [4:0]wb_rfaddr_out,
    output [31:0]wb_rf_out,
    output wb_rf_allow,
    output [31:0]wb_hi_out,

```

```
    output [31:0]wb_lo_out
);
```

四、实验仿真过程

1、动态流水线的仿真过程

1.1 Testbench 程序

```
'timescale 1ns / 1ps

module cpu_tb();
    reg clk;
    reg rst;
    wire [7:0] o_seg;
    wire [7:0] o_sel;

    //看控制信号
    wire [31:0]id_pc_in = uut.instdecode.id_pc_in;
    wire [31:0]id_inst_in = uut.instdecode.id_inst_in;
    wire [31:0]stage_id_inst = uut.stage_id_inst;
    wire [31:0]stage_exe_inst = uut.stage_exe_inst;

    wire isbranch = uut.isbranch;
    wire [31:0]branch_pc = uut.branch_pc;

    wire [1:0]stage_exe_rfaddrinchoice = uut.instdecode.stage_exe_rfaddrinchoice;

    wire [31:0]id_rs = uut.instdecode.id_rs;
    wire [31:0]id_rt = uut.instdecode.id_rt;

    wire [31:0]id_alu_a = uut.instdecode.id_alu_a;
    wire [31:0]id_alu_b = uut.instdecode.id_alu_b;
    wire [4:0]id_aluchoice = uut.instdecode.id_aluchoice;
    wire [31:0]exe_alu_out = uut.exe_alu_out;
    wire [31:0]exe_wb_hiin = uut.exe_wb_hiin;
    wire [31:0]exe_wb_loin = uut.exe_wb_loin;

    wire [4:0]wb_rfaddr_out = uut.wb_rfaddr_out;
    wire [31:0]wb_rf_out = uut.wb_rf_out;
    wire wb_rf_allow_out = uut.wb_rf_allow_out;
```

```

wire [31:0]mem_dmem_out = uut.memory.mem_dmem_out;
wire [31:0]mem_dmem_addr = uut.memory.mem_dmem_addr;
wire [31:0]mem_dmem_in = uut.memory.mem_dmem_in;

wire [31:0]ans = uut.ans;

wire [1:0]ischangea = uut.instdecode.ischangea;
wire [1:0]ischangeb = uut.instdecode.ischangeb;
wire clk_in = uut.clk_in;

sccomp_dataflow uut(
    clk,
    rst,
    o_seg,
    o_sel
);

integer file_output;
integer counter;
initial
begin
    file_output = $fopen("D:/cputest/my/my_1.1_d.txt");//可根据需要调整
    clk = 1;
    rst = 1;
    counter = 0;
    #4;
    rst = 0;
    counter = 0;
end

always
begin
    #2;
    clk = ~clk;
    if (clk == 1'b1)
        begin
            if (counter == 4000)
                begin
                    $fclose(file_output);
                end
            else
                begin
                    counter = counter + 1;
                    begin

```

```
$fdisplay(file_output,"pc: %h",uut.stage_if_pc);
$fdisplay(file_output,"instr: %h",uut.stage_if_inst);

$fdisplay(file_output,"regfile0: %h",uut.instdecode.cpu_ref.array_reg[0]);

$fdisplay(file_output,"regfile1: %h",uut.instdecode.cpu_ref.array_reg[1]);

$fdisplay(file_output,"regfile2: %h",uut.instdecode.cpu_ref.array_reg[2]);

$fdisplay(file_output,"regfile3: %h",uut.instdecode.cpu_ref.array_reg[3]);

$fdisplay(file_output,"regfile4: %h",uut.instdecode.cpu_ref.array_reg[4]);

$fdisplay(file_output,"regfile5: %h",uut.instdecode.cpu_ref.array_reg[5]);

$fdisplay(file_output,"regfile6: %h",uut.instdecode.cpu_ref.array_reg[6]);

$fdisplay(file_output,"regfile7: %h",uut.instdecode.cpu_ref.array_reg[7]);

$fdisplay(file_output,"regfile8: %h",uut.instdecode.cpu_ref.array_reg[8]);

$fdisplay(file_output,"regfile9: %h",uut.instdecode.cpu_ref.array_reg[9]);

$fdisplay(file_output,"regfile10: %h",uut.instdecode.cpu_ref.array_reg[10]);

$fdisplay(file_output,"regfile11: %h",uut.instdecode.cpu_ref.array_reg[11]);

$fdisplay(file_output,"regfile12: %h",uut.instdecode.cpu_ref.array_reg[12]);

$fdisplay(file_output,"regfile13: %h",uut.instdecode.cpu_ref.array_reg[13]);

$fdisplay(file_output,"regfile14: %h",uut.instdecode.cpu_ref.array_reg[14]);

$fdisplay(file_output,"regfile15: %h",uut.instdecode.cpu_ref.array_reg[15]);

$fdisplay(file_output,"regfile16: %h",uut.instdecode.cpu_ref.array_reg[16]);

$fdisplay(file_output,"regfile17: %h",uut.instdecode.cpu_ref.array_reg[17]);

$fdisplay(file_output,"regfile18: %h",uut.instdecode.cpu_ref.array_reg[18]);

$fdisplay(file_output,"regfile19: %h",uut.instdecode.cpu_ref.array_reg[19]);

$fdisplay(file_output,"regfile20: %h",uut.instdecode.cpu_ref.array_reg[20]);
```

```

$fdisplay(file_output,"regfile21: %h",uut.instdecode.cpu_ref.array_reg[21]);

$fdisplay(file_output,"regfile22: %h",uut.instdecode.cpu_ref.array_reg[22]);

$fdisplay(file_output,"regfile23: %h",uut.instdecode.cpu_ref.array_reg[23]);

$fdisplay(file_output,"regfile24: %h",uut.instdecode.cpu_ref.array_reg[24]);

$fdisplay(file_output,"regfile25: %h",uut.instdecode.cpu_ref.array_reg[25]);

$fdisplay(file_output,"regfile26: %h",uut.instdecode.cpu_ref.array_reg[26]);

$fdisplay(file_output,"regfile27: %h",uut.instdecode.cpu_ref.array_reg[27]);

$fdisplay(file_output,"regfile28: %h",uut.instdecode.cpu_ref.array_reg[28]);

$fdisplay(file_output,"regfile29: %h",uut.instdecode.cpu_ref.array_reg[29]);

$fdisplay(file_output,"regfile30: %h",uut.instdecode.cpu_ref.array_reg[30]);

$fdisplay(file_output,"regfile31: %h",uut.instdecode.cpu_ref.array_reg[31]);
    end
  end
end
endmodule

```

1.2 测试情况

在本次流水线实验中，我采用之前计算机组成原理设计课上提供的 54 条指令 CPU 的测试文件，将 pc、指令以及寄存器的值输出到 txt 文件中进行了测试；由于流水线 CPU 无法在下一条指令取指前写回寄存器，因此只能通过观察最终的寄存器情况来判定 CPU 设计是否正确。

ADDI:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_1_addi.result.txt D:\cputest\my\my_1_addi.txt

```

1941 regfile0: 00000000
1942 regfile1: 00000001
1943 regfile2: ffff8000
1944 regfile3: 00007fff
1945 regfile4: 00000000
1946 regfile5: ffff7fff
1947 regfile6: 00008005
1948 regfile7: 0000000f
1949 regfile8: 000000f0
1950 regfile9: 00000f00
1951 regfile10: fffff000
1952 regfile11: 0000aaaa
1953 regfile12: fffff554
1954 regfile13: 0000001e
1955 regfile14: 000001fe
1956 regfile15: 00001ffe
1957 regfile16: 00000011
1958 regfile17: fffff012
1959 regfile18: fffff8013
1960 regfile19: 00007014
1961 regfile20: 00000015
1962 regfile21: 00000226
1963 regfile22: 00000ac7
1964 regfile23: fffffdc18
1965 regfile24: fffffa019
1966 regfile25: fffffb01a
1967 regfile26: fffffe01b
1968 regfile27: 00007d1c
1969 regfile28: fffffcbbd
1970 regfile29: fffffdbde
1971 regfile30: 0000788f
1972 regfile31: 00007790

3369 regfile0: 00000000
3370 regfile1: 00000001
3371 regfile2: ffff8000
3372 regfile3: 00007fff
3373 regfile4: 00000000
3374 regfile5: ffff7fff
3375 regfile6: 00008005
3376 regfile7: 0000000f
3377 regfile8: 000000f0
3378 regfile9: 00000f00
3379 regfile10: fffff000
3380 regfile11: 0000aaaa
3381 regfile12: fffff554
3382 regfile13: 0000001e
3383 regfile14: 000001fe
3384 regfile15: 00001ffe
3385 regfile16: 00000011
3386 regfile17: fffff012
3387 regfile18: fffff8013
3388 regfile19: 00007014
3389 regfile20: 00000015
3390 regfile21: 00000226
3391 regfile22: 00000ac7
3392 regfile23: fffffdc18
3393 regfile24: fffffa019
3394 regfile25: fffffb01a
3395 regfile26: fffffe01b
3396 regfile27: 00007d1c
3397 regfile28: fffffcbbd
3398 regfile29: fffffdbde
3399 regfile30: 0000788f
3400 regfile31: 00007790

```

+ - 1868 lines unchanged, 1428 lines added, 104 lines modified, 0 lines deleted.

ADDIU:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_2_addiu.result.txt D:\cputest\my\my_2_addiu.txt

```

1941 regfile0: 00000000
1942 regfile1: 00000001
1943 regfile2: ffff8000
1944 regfile3: 00007fff
1945 regfile4: 00000000
1946 regfile5: ffff7fff
1947 regfile6: 00008005
1948 regfile7: 0000000f
1949 regfile8: 000000f0
1950 regfile9: 00000f00
1951 regfile10: fffff000
1952 regfile11: 0000aaaa
1953 regfile12: fffff554
1954 regfile13: 0000001e
1955 regfile14: 000001fe
1956 regfile15: 00001ffe
1957 regfile16: 00000011
1958 regfile17: fffff012
1959 regfile18: fffff8013
1960 regfile19: 00007014
1961 regfile20: 00000015
1962 regfile21: 00000226
1963 regfile22: 00000ac7
1964 regfile23: fffffdc18
1965 regfile24: fffffa019
1966 regfile25: fffffb01a
1967 regfile26: fffffe01b
1968 regfile27: 00007d1c
1969 regfile28: fffffcbbd
1970 regfile29: fffffdbde
1971 regfile30: 0000788f
1972 regfile31: 00007790

3369 regfile0: 00000000
3370 regfile1: 00000001
3371 regfile2: ffff8000
3372 regfile3: 00007fff
3373 regfile4: 00000000
3374 regfile5: ffff7fff
3375 regfile6: 00008005
3376 regfile7: 0000000f
3377 regfile8: 000000f0
3378 regfile9: 00000f00
3379 regfile10: fffff000
3380 regfile11: 0000aaaa
3381 regfile12: fffff554
3382 regfile13: 0000001e
3383 regfile14: 000001fe
3384 regfile15: 00001ffe
3385 regfile16: 00000011
3386 regfile17: fffff012
3387 regfile18: fffff8013
3388 regfile19: 00007014
3389 regfile20: 00000015
3390 regfile21: 00000226
3391 regfile22: 00000ac7
3392 regfile23: fffffdc18
3393 regfile24: fffffa019
3394 regfile25: fffffb01a
3395 regfile26: fffffe01b
3396 regfile27: 00007d1c
3397 regfile28: fffffcbbd
3398 regfile29: fffffdbde
3399 regfile30: 0000788f
3400 regfile31: 00007790

```

+ - 1868 lines unchanged, 1428 lines added, 104 lines modified, 0 lines deleted.

ANDI:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_3_andi.txt

```

2145 regfile0: 00000000
2146 regfile1: 00000000
2147 regfile2: 00000000
2148 regfile3: 00000000
2149 regfile4: 00000000
2150 regfile5: 00000000
2151 regfile6: 00000000
2152 regfile7: 00000000
2153 regfile8: 00000000
2154 regfile9: 0000ffff
2155 regfile10: 00008000
2156 regfile11: 00007fff
2157 regfile12: 0000f0f0
2158 regfile13: 00005555
2159 regfile14: 0000aaaa
2160 regfile15: 00001234
2161 regfile16: 00000000
2162 regfile17: 0000ffff
2163 regfile18: 00008000
2164 regfile19: 00007fff
2165 regfile20: 0000f0f0
2166 regfile21: 00005555
2167 regfile22: 0000aaaa
2168 regfile23: 00001234
2169 regfile24: 00000000
2170 regfile25: 00000000
2171 regfile26: 00000000
2172 regfile27: 00000000
2173 regfile28: 00000000
2174 regfile29: 00000000
2175 regfile30: 00000000
2176 regfile31: 00000000

```

D:\cputest\my\my_3_andi.txt

```

3369 regfile0: 00000000
3370 regfile1: 00000000
3371 regfile2: 00000000
3372 regfile3: 00000000
3373 regfile4: 00000000
3374 regfile5: 00000000
3375 regfile6: 00000000
3376 regfile7: 00000000
3377 regfile8: 00000000
3378 regfile9: 0000ffff
3379 regfile10: 00008000
3380 regfile11: 00007fff
3381 regfile12: 0000f0f0
3382 regfile13: 00005555
3383 regfile14: 0000aaaa
3384 regfile15: 00001234
3385 regfile16: 00000000
3386 regfile17: 0000ffff
3387 regfile18: 00008000
3388 regfile19: 00007fff
3389 regfile20: 0000f0f0
3390 regfile21: 00005555
3391 regfile22: 0000aaaa
3392 regfile23: 00001234
3393 regfile24: 00000000
3394 regfile25: 00000000
3395 regfile26: 00000000
3396 regfile27: 00000000
3397 regfile28: 00000000
3398 regfile29: 00000000
3399 regfile30: 00000000
3400 regfile31: 00000000

```

+ - 2069 lines unchanged, 1224 lines added, 107 lines modified, 0 lines deleted.

ORI:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_4_ori.txt

```

2145 regfile0: 00000000
2146 regfile1: 0000ffff
2147 regfile2: 00008000
2148 regfile3: 00007fff
2149 regfile4: 0000f0f0
2150 regfile5: 00005555
2151 regfile6: 0000aaaa
2152 regfile7: 00001234
2153 regfile8: 00000000
2154 regfile9: fffffff
2155 regfile10: fffffff
2156 regfile11: fffffff
2157 regfile12: fffffff
2158 regfile13: fffffff
2159 regfile14: fffffff
2160 regfile15: fffffff
2161 regfile16: fffffff
2162 regfile17: fffffff
2163 regfile18: ffff8000
2164 regfile19: 00007fff
2165 regfile20: fffff0f0
2166 regfile21: 00005555
2167 regfile22: fffffaaa
2168 regfile23: 00001234
2169 regfile24: 00000000
2170 regfile25: 0000ffff
2171 regfile26: 0000ffff
2172 regfile27: 0000ffff
2173 regfile28: 0000ffff
2174 regfile29: 0000ffff
2175 regfile30: 0000ffff
2176 regfile31: ffffffff

```

D:\cputest\my\my_4_ori.txt

```

3369 regfile0: 00000000
3370 regfile1: 0000ffff
3371 regfile2: 00008000
3372 regfile3: 00007fff
3373 regfile4: 0000f0f0
3374 regfile5: 00005555
3375 regfile6: 0000aaaa
3376 regfile7: 00001234
3377 regfile8: 00000000
3378 regfile9: fffffff
3379 regfile10: fffffff
3380 regfile11: fffffff
3381 regfile12: fffffff
3382 regfile13: fffffff
3383 regfile14: fffffff
3384 regfile15: fffffff
3385 regfile16: fffffff
3386 regfile17: fffffff
3387 regfile18: ffff8000
3388 regfile19: 00007fff
3389 regfile20: fffff0f0
3390 regfile21: 00005555
3391 regfile22: fffffaaa
3392 regfile23: 00001234
3393 regfile24: 00000000
3394 regfile25: 0000ffff
3395 regfile26: 0000ffff
3396 regfile27: 0000ffff
3397 regfile28: 0000ffff
3398 regfile29: 0000ffff
3399 regfile30: 0000ffff
3400 regfile31: ffffffff

```

+ - 2069 lines unchanged, 1224 lines added, 107 lines modified, 0 lines deleted.

SLTIU:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_5_situ.txt

```

2145 regfile0: 00000000
2146 regfile1: 00000001
2147 regfile2: 00000001
2148 regfile3: 00000001
2149 regfile4: 00000001
2150 regfile5: 00000000
2151 regfile6: 00000000
2152 regfile7: 00000000
2153 regfile8: 00000000
2154 regfile9: 00000000
2155 regfile10: 00000000
2156 regfile11: 00000000
2157 regfile12: 00000000
2158 regfile13: 00000000
2159 regfile14: 00000000
2160 regfile15: 00000000
2161 regfile16: 00000001
2162 regfile17: 00000001
2163 regfile18: 00000001
2164 regfile19: 00000001
2165 regfile20: 00000001
2166 regfile21: 00000000
2167 regfile22: 00000001
2168 regfile23: 00000001
2169 regfile24: 00000001
2170 regfile25: 00000001
2171 regfile26: 00000001
2172 regfile27: 00000001
2173 regfile28: 00000000
2174 regfile29: 00000001
2175 regfile30: 00000001
2176 regfile31: 00000001

```

D:\cputest\my\my_5_situ.txt

```

3369 regfile0: 00000000
3370 regfile1: 00000001
3371 regfile2: 00000001
3372 regfile3: 00000001
3373 regfile4: 00000001
3374 regfile5: 00000000
3375 regfile6: 00000000
3376 regfile7: 00000000
3377 regfile8: 00000000
3378 regfile9: 00000000
3379 regfile10: 00000000
3380 regfile11: 00000000
3381 regfile12: 00000000
3382 regfile13: 00000000
3383 regfile14: 00000000
3384 regfile15: 00000000
3385 regfile16: 00000001
3386 regfile17: 00000001
3387 regfile18: 00000001
3388 regfile19: 00000001
3389 regfile20: 00000001
3390 regfile21: 00000000
3391 regfile22: 00000001
3392 regfile23: 00000001
3393 regfile24: 00000001
3394 regfile25: 00000001
3395 regfile26: 00000001
3396 regfile27: 00000001
3397 regfile28: 00000000
3398 regfile29: 00000001
3399 regfile30: 00000001
3400 regfile31: 00000001

```

+ - 2057 lines unchanged, 1224 lines added, 119 lines modified, 0 lines deleted.

LUI:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_6_lui.txt

```

1091 regfile0: 00000000
1092 regfile1: 00010000
1093 regfile2: 80000000
1094 regfile3: ffff0000
1095 regfile4: abcf0000
1096 regfile5: llff0000
1097 regfile6: fd3f0000
1098 regfile7: faff0000
1099 regfile8: 01ff0000
1100 regfile9: 45ff0000
1101 regfile10: f00f0000
1102 regfile11: offf0000
1103 regfile12: dfff0000
1104 regfile13: efff0000
1105 regfile14: bfff0000
1106 regfile15: ff010000
1107 regfile16: f00f0000
1108 regfile17: f00f0000
1109 regfile18: f00f0000
1110 regfile19: f00f0000
1111 regfile20: 00000000
1112 regfile21: f00f0000
1113 regfile22: f00f0000
1114 regfile23: f00f0000
1115 regfile24: f00f0000
1116 regfile25: f00f0000
1117 regfile26: f00f0000
1118 regfile27: f00f0000
1119 regfile28: f00f0000
1120 regfile29: 00000000
1121 regfile30: 00000000
1122 regfile31: f00f0000

```

D:\cputest\my\my_6_lui.txt

```

3369 regfile0: 00000000
3370 regfile1: 00010000
3371 regfile2: 80000000
3372 regfile3: ffff0000
3373 regfile4: abcf0000
3374 regfile5: llff0000
3375 regfile6: fd3f0000
3376 regfile7: faff0000
3377 regfile8: 01ff0000
3378 regfile9: 45ff0000
3379 regfile10: f00f0000
3380 regfile11: offf0000
3381 regfile12: dfff0000
3382 regfile13: efff0000
3383 regfile14: bfff0000
3384 regfile15: ff010000
3385 regfile16: f00f0000
3386 regfile17: f00f0000
3387 regfile18: f00f0000
3388 regfile19: f00f0000
3389 regfile20: 00000000
3390 regfile21: f00f0000
3391 regfile22: f00f0000
3392 regfile23: f00f0000
3393 regfile24: f00f0000
3394 regfile25: f00f0000
3395 regfile26: f00f0000
3396 regfile27: f00f0000
3397 regfile28: f00f0000
3398 regfile29: 00000000
3399 regfile30: 00000000
3400 regfile31: f00f0000

```

+ - 1063 lines unchanged, 2278 lines added, 59 lines modified, 0 lines deleted.

OXR:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_7_xori.result.txt D:\cputest\my\my_7_xor.txt

```

2145 regfile0: 00000000
2146 regfile1: 0000ffff
2147 regfile2: 00008000
2148 regfile3: 00007fff
2149 regfile4: 0000f0f0
2150 regfile5: 00005555
2151 regfile6: 0000aaaa
2152 regfile7: 00001234
2153 regfile8: 00000000
2154 regfile9: ffff0000
2155 regfile10: ffffffff
2156 regfile11: ffff8000
2157 regfile12: ffff0f0f
2158 regfile13: fffffaaa
2159 regfile14: fffff555
2160 regfile15: fffffedcb
2161 regfile16: ffffffff
2162 regfile17: fffff000
2163 regfile18: fffff000
2164 regfile19: 00000000
2165 regfile20: ffff0000
2166 regfile21: 00000000
2167 regfile22: ffff0000
2168 regfile23: 00000000
2169 regfile24: 00000000
2170 regfile25: 0000ffff
2171 regfile26: 0000ffff
2172 regfile27: ffffffff
2173 regfile28: 0000ffff
2174 regfile29: ffffffff
2175 regfile30: 0000ffff
2176 regfile31: ffffffff

```

3369 regfile0: 00000000
3370 regfile1: 0000ffff
3371 regfile2: 00008000
3372 regfile3: 00007fff
3373 regfile4: 0000f0f0
3374 regfile5: 00005555
3375 regfile6: 0000aaaa
3376 regfile7: 00001234
3377 regfile8: 00000000
3378 regfile9: ffff0000
3379 regfile10: fffff7fff
3380 regfile11: ffff8000
3381 regfile12: ffff0f0f
3382 regfile13: fffffaaa
3383 regfile14: fffff555
3384 regfile15: fffffedcb
3385 regfile16: ffffffff
3386 regfile17: fffff000
3387 regfile18: fffff000
3388 regfile19: 00000000
3389 regfile20: ffff0000
3390 regfile21: 00000000
3391 regfile22: ffff0000
3392 regfile23: 00000000
3393 regfile24: 00000000
3394 regfile25: 0000ffff
3395 regfile26: 0000ffff
3396 regfile27: ffffffff
3397 regfile28: 0000ffff
3398 regfile29: ffffffff
3399 regfile30: 0000ffff
3400 regfile31: ffffffff

2069 lines unchanged, 1224 lines added, 107 lines modified, 0 lines deleted.

SLTI:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_8_slti.result.txt D:\cputest\my\my_8_slti.txt

```

2145 regfile0: 00000000
2146 regfile1: 00000000
2147 regfile2: 00000001
2148 regfile3: 00000000
2149 regfile4: 00000001
2150 regfile5: 00000000
2151 regfile6: 00000001
2152 regfile7: 00000000
2153 regfile8: 00000000
2154 regfile9: 00000001
2155 regfile10: 00000000
2156 regfile11: 00000001
2157 regfile12: 00000000
2158 regfile13: 00000001
2159 regfile14: 00000001
2160 regfile15: 00000000
2161 regfile16: 00000001
2162 regfile17: 00000000
2163 regfile18: 00000001
2164 regfile19: 00000000
2165 regfile20: 00000001
2166 regfile21: 00000000
2167 regfile22: 00000000
2168 regfile23: 00000001
2169 regfile24: 00000000
2170 regfile25: 00000001
2171 regfile26: 00000000
2172 regfile27: 00000001
2173 regfile28: 00000000
2174 regfile29: 00000000
2175 regfile30: 00000001
2176 regfile31: 00000000

```

3369 regfile0: 00000000
3370 regfile1: 00000000
3371 regfile2: 00000001
3372 regfile3: 00000000
3373 regfile4: 00000001
3374 regfile5: 00000000
3375 regfile6: 00000001
3376 regfile7: 00000000
3377 regfile8: 00000000
3378 regfile9: 00000000
3379 regfile10: 00000000
3380 regfile11: 00000001
3381 regfile12: 00000000
3382 regfile13: 00000001
3383 regfile14: 00000001
3384 regfile15: 00000000
3385 regfile16: 00000001
3386 regfile17: 00000000
3387 regfile18: 00000001
3388 regfile19: 00000000
3389 regfile20: 00000001
3390 regfile21: 00000000
3391 regfile22: 00000000
3392 regfile23: 00000001
3393 regfile24: 00000000
3394 regfile25: 00000001
3395 regfile26: 00000000
3396 regfile27: 00000001
3397 regfile28: 00000000
3398 regfile29: 00000000
3399 regfile30: 00000001
3400 regfile31: 00000000

2057 lines unchanged, 1224 lines added, 119 lines modified, 0 lines deleted.

ADDU:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_9_addu.result.txt D:\cputest\my\my_9_addu.txt

```

5035 regfile0: 00000000
5036 regfile1: 00000001
5037 regfile2: ffffffff
5038 regfile3: 80000000
5039 regfile4: 80000000
5040 regfile5: 80000001
5041 regfile6: 80000002
5042 regfile7: 80000004
5043 regfile8: 80000004
5044 regfile9: 80000005
5045 regfile10: 80000006
5046 regfile11: 80000007
5047 regfile12: 80000008
5048 regfile13: 80000009
5049 regfile14: 8000000a
5050 regfile15: 8000000b
5051 regfile16: 8000000c
5052 regfile17: 8000000d
5053 regfile18: 8000000e
5054 regfile19: 70000000
5055 regfile20: 7fff8000
5056 regfile21: 00000000
5057 regfile22: 7fffffff
5058 regfile23: 7fffff0
5059 regfile24: 0fff0000
5060 regfile25: 1fff0000
5061 regfile26: 6fff0000
5062 regfile27: 0f300000
5063 regfile28: 001c0000
5064 regfile29: 001d0000
5065 regfile30: 001e0000
5066 regfile31: 001f0000
16969 regfile0: 00000000
16970 regfile1: 00000001
16971 regfile2: ffffffff
16972 regfile3: 80000000
16973 regfile4: 80000000
16974 regfile5: 80000001
16975 regfile6: 80000002
16976 regfile7: 80000004
16977 regfile8: 80000004
16978 regfile9: 80000005
16979 regfile10: 80000006
16980 regfile11: 80000007
16981 regfile12: 80000008
16982 regfile13: 80000009
16983 regfile14: 8000000a
16984 regfile15: 8000000b
16985 regfile16: 8000000c
16986 regfile17: 8000000d
16987 regfile18: 8000000e
16988 regfile19: 70000000
16989 regfile20: 7fff8000
16990 regfile21: 00000000
16991 regfile22: 7fffffff
16992 regfile23: 7fffff0
16993 regfile24: 0fff0000
16994 regfile25: 1fff0000
16995 regfile26: 6fff0000
16996 regfile27: 0f300000
16997 regfile28: 001c0000
16998 regfile29: 001d0000
16999 regfile30: 001e0000
17000 regfile31: 001f0000

```

+ F 4910 lines unchanged, 11934 lines added, 156 lines modified, 0 lines deleted.

AND:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_10_and.result.txt D:\cputest\my\my_10_and.txt

```

5103 regfile0: 00000000
5104 regfile1: 00000001
5105 regfile2: ffffffff
5106 regfile3: 80000000
5107 regfile4: 00005556
5108 regfile5: ffffaaab
5109 regfile6: 00000000
5110 regfile7: 00000000
5111 regfile8: 00000000
5112 regfile9: 00000000
5113 regfile10: 00000000
5114 regfile11: 00000000
5115 regfile12: 00000000
5116 regfile13: 00000000
5117 regfile14: 00000000
5118 regfile15: 00000000
5119 regfile16: 00000000
5120 regfile17: 00000000
5121 regfile18: 00000000
5122 regfile19: 00000000
5123 regfile20: 80000000
5124 regfile21: 00000000
5125 regfile22: 00000000
5126 regfile23: 00000000
5127 regfile24: 00000000
5128 regfile25: 00000000
5129 regfile26: 00000000
5130 regfile27: 00000000
5131 regfile28: 00000000
5132 regfile29: 00000000
5133 regfile30: 99910000
5134 regfile31: 74650000
16969 regfile0: 00000000
16970 regfile1: 00000001
16971 regfile2: ffffffff
16972 regfile3: 80000000
16973 regfile4: 00005556
16974 regfile5: ffffaaab
16975 regfile6: 00000000
16976 regfile7: 00000000
16977 regfile8: 00000000
16978 regfile9: 00000000
16979 regfile10: 00000000
16980 regfile11: 00000000
16981 regfile12: 00000000
16982 regfile13: 00000000
16983 regfile14: 00000000
16984 regfile15: 00000000
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 00000000
16988 regfile19: 00000000
16989 regfile20: 80000000
16990 regfile21: 00000000
16991 regfile22: 00000000
16992 regfile23: 00000000
16993 regfile24: 00000000
16994 regfile25: 00000000
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 00000000
16999 regfile30: 99910000
17000 regfile31: 74650000

```

+ F 5003 lines unchanged, 11866 lines added, 131 lines modified, 0 lines deleted.

BEQ:

TextDiff

File Edit Search Options Actions Help

```
D:\cputest\my\my_11_beq.result.txt          D:\cputest\my\my_11_beq.txt
1975 regfile0: 00000000
1976 regfile1: ffffffff
1977 regfile2: ffffffff
1978 regfile3: ffffffff
1979 regfile4: ffffffff
1980 regfile5: ffffffff
1981 regfile6: ffffffff
1982 regfile7: ffffffff
1983 regfile8: ffffffff
1984 regfile9: ffffffff
1985 regfile10: ffffffff
1986 regfile11: ffffffff
1987 regfile12: ffffffff
1988 regfile13: ffffffff
1989 regfile14: ffffffff
1990 regfile15: ffffffff
1991 regfile16: ffffffff
1992 regfile17: ffffffff
1993 regfile18: ffffffff
1994 regfile19: ffffffff
1995 regfile20: ffffffff
1996 regfile21: ffffffff
1997 regfile22: ffffffff
1998 regfile23: ffffffff
1999 regfile24: ffffffff
2000 regfile25: ffffffff
2001 regfile26: ffffffff
2002 regfile27: ffffffff
2003 regfile28: ffffffff
2004 regfile29: ffffffff
2005 regfile30: ffffffff
2006 regfile31: ffffffff

16969 regfile0: 00000000
16970 regfile1: ffffffff
16971 regfile2: ffffffff
16972 regfile3: ffffffff
16973 regfile4: ffffffff
16974 regfile5: ffffffff
16975 regfile6: ffffffff
16976 regfile7: ffffffff
16977 regfile8: ffffffff
16978 regfile9: ffffffff
16979 regfile10: ffffffff
16980 regfile11: ffffffff
16981 regfile12: ffffffff
16982 regfile13: ffffffff
16983 regfile14: ffffffff
16984 regfile15: ffffffff
16985 regfile16: ffffffff
16986 regfile17: ffffffff
16987 regfile18: ffffffff
16988 regfile19: ffffffff
16989 regfile20: ffffffff
16990 regfile21: ffffffff
16991 regfile22: ffffffff
16992 regfile23: ffffffff
16993 regfile24: ffffffff
16994 regfile25: ffffffff
16995 regfile26: ffffffff
16996 regfile27: ffffffff
16997 regfile28: ffffffff
16998 regfile29: ffffffff
16999 regfile30: ffffffff
17000 regfile31: ffffffff

< > < > + F 1476 lines unchanged, 14994 lines added, 530 lines modified, 0 lines deleted.
```

BNE:

TextDiff

File Edit Search Options Actions Help

```
D:\cputest\my\my_12_bne.result.txt          D:\cputest\my\my_12_bne.txt
2519 regfile0: 00000000
2520 regfile1: ffffffff
2521 regfile2: ffffffff
2522 regfile3: ffffffff
2523 regfile4: ffffffff
2524 regfile5: ffffffff
2525 regfile6: ffffffff
2526 regfile7: ffffffff
2527 regfile8: ffffffff
2528 regfile9: ffffffff
2529 regfile10: ffffffff
2530 regfile11: ffffffff
2531 regfile12: ffffffff
2532 regfile13: ffffffff
2533 regfile14: ffffffff
2534 regfile15: ffffffff
2535 regfile16: ffffffff
2536 regfile17: ffffffff
2537 regfile18: ffffffff
2538 regfile19: ffffffff
2539 regfile20: ffffffff
2540 regfile21: ffffffff
2541 regfile22: ffffffff
2542 regfile23: ffffffff
2543 regfile24: ffffffff
2544 regfile25: ffffffff
2545 regfile26: ffffffff
2546 regfile27: ffffffff
2547 regfile28: ffffffff
2548 regfile29: ffffffff
2549 regfile30: ffffffff
2550 regfile31: ffffffff

16969 regfile0: 00000000
16970 regfile1: ffffffff
16971 regfile2: ffffffff
16972 regfile3: ffffffff
16973 regfile4: ffffffff
16974 regfile5: ffffffff
16975 regfile6: ffffffff
16976 regfile7: ffffffff
16977 regfile8: ffffffff
16978 regfile9: ffffffff
16979 regfile10: ffffffff
16980 regfile11: ffffffff
16981 regfile12: ffffffff
16982 regfile13: ffffffff
16983 regfile14: ffffffff
16984 regfile15: ffffffff
16985 regfile16: ffffffff
16986 regfile17: ffffffff
16987 regfile18: ffffffff
16988 regfile19: ffffffff
16989 regfile20: ffffffff
16990 regfile21: ffffffff
16991 regfile22: ffffffff
16992 regfile23: ffffffff
16993 regfile24: ffffffff
16994 regfile25: ffffffff
16995 regfile26: ffffffff
16996 regfile27: ffffffff
16997 regfile28: ffffffff
16998 regfile29: ffffffff
16999 regfile30: ffffffff
17000 regfile31: ffffffff

< > < > + F 2453 lines unchanged, 14450 lines added, 97 lines modified, 0 lines deleted.
```

J:

TextDiff

File Edit Search Options Actions Help

D:\cpustest\my\my_13_j.txt D:\cpustest\my\my_13_j.txt

```

3335 regfile0: 00000000
3336 regfile1: ffffffff
3337 regfile2: ffffffff
3338 regfile3: ffffffff
3339 regfile4: ffffffff
3340 regfile5: ffffffff
3341 regfile6: ffffffff
3342 regfile7: ffffffff
3343 regfile8: ffffffff
3344 regfile9: ffffffff
3345 regfile10: ffffffff
3346 regfile11: ffffffff
3347 regfile12: ffffffff
3348 regfile13: ffffffff
3349 regfile14: ffffffff
3350 regfile15: ffffffff
3351 regfile16: ffffffff
3352 regfile17: ffffffff
3353 regfile18: ffffffff
3354 regfile19: ffffffff
3355 regfile20: ffffffff
3356 regfile21: ffffffff
3357 regfile22: ffffffff
3358 regfile23: ffffffff
3359 regfile24: ffffffff
3360 regfile25: ffffffff
3361 regfile26: ffffffff
3362 regfile27: ffffffff
3363 regfile28: ffffffff
3364 regfile29: ffffffff
3365 regfile30: ffffffff
3366 regfile31: ffffffff

16969 regfile0: 00000000
16970 regfile1: ffffffff
16971 regfile2: ffffffff
16972 regfile3: ffffffff
16973 regfile4: ffffffff
16974 regfile5: ffffffff
16975 regfile6: ffffffff
16976 regfile7: ffffffff
16977 regfile8: ffffffff
16978 regfile9: ffffffff
16979 regfile10: ffffffff
16980 regfile11: ffffffff
16981 regfile12: ffffffff
16982 regfile13: ffffffff
16983 regfile14: ffffffff
16984 regfile15: ffffffff
16985 regfile16: ffffffff
16986 regfile17: ffffffff
16987 regfile18: ffffffff
16988 regfile19: ffffffff
16989 regfile20: ffffffff
16990 regfile21: ffffffff
16991 regfile22: ffffffff
16992 regfile23: ffffffff
16993 regfile24: ffffffff
16994 regfile25: ffffffff
16995 regfile26: ffffffff
16996 regfile27: ffffffff
16997 regfile28: ffffffff
16998 regfile29: ffffffff
16999 regfile30: ffffffff
17000 regfile31: ffffffff

```

+ F 3177 lines unchanged, 13634 lines added, 189 lines modified, 0 lines deleted.

JAL:

TextDiff

File Edit Search Options Actions Help

D:\cpustest\my\my_14_jal.txt D:\cpustest\my\my_14_jal.txt

```

3267 regfile0: 00000000
3268 regfile1: ffffffff
3269 regfile2: ffffffff
3270 regfile3: ffffffff
3271 regfile4: ffffffff
3272 regfile5: ffffffff
3273 regfile6: ffffffff
3274 regfile7: ffffffff
3275 regfile8: ffffffff
3276 regfile9: ffffffff
3277 regfile10: ffffffff
3278 regfile11: ffffffff
3279 regfile12: ffffffff
3280 regfile13: ffffffff
3281 regfile14: ffffffff
3282 regfile15: ffffffff
3283 regfile16: ffffffff
3284 regfile17: ffffffff
3285 regfile18: ffffffff
3286 regfile19: ffffffff
3287 regfile20: ffffffff
3288 regfile21: ffffffff
3289 regfile22: ffffffff
3290 regfile23: ffffffff
3291 regfile24: ffffffff
3292 regfile25: ffffffff
3293 regfile26: ffffffff
3294 regfile27: ffffffff
3295 regfile28: ffffffff
3296 regfile29: ffffffff
3297 regfile30: ffffffff
3298 regfile31: ffffffff

16969 regfile0: 00000000
16970 regfile1: ffffffff
16971 regfile2: ffffffff
16972 regfile3: ffffffff
16973 regfile4: ffffffff
16974 regfile5: ffffffff
16975 regfile6: ffffffff
16976 regfile7: ffffffff
16977 regfile8: ffffffff
16978 regfile9: ffffffff
16979 regfile10: ffffffff
16980 regfile11: ffffffff
16981 regfile12: ffffffff
16982 regfile13: ffffffff
16983 regfile14: ffffffff
16984 regfile15: ffffffff
16985 regfile16: ffffffff
16986 regfile17: ffffffff
16987 regfile18: ffffffff
16988 regfile19: ffffffff
16989 regfile20: ffffffff
16990 regfile21: ffffffff
16991 regfile22: ffffffff
16992 regfile23: ffffffff
16993 regfile24: ffffffff
16994 regfile25: ffffffff
16995 regfile26: ffffffff
16996 regfile27: ffffffff
16997 regfile28: ffffffff
16998 regfile29: ffffffff
16999 regfile30: ffffffff
17000 regfile31: ffffffff

```

+ F 3049 lines unchanged, 13702 lines added, 249 lines modified, 0 lines deleted.

JR:

TextDiff

File Edit Search Options Actions Help

D:\cpustest\my\my_15_jr.txt D:\cpustest\my\my_15_jr.txt

```

615 regfile0: 00000000
616 regfile1: 00000040
617 regfile2: 00000000
618 regfile3: 00000000
619 regfile4: 00000000
620 regfile5: ffffffff
621 regfile6: 00000001
622 regfile7: 00000003
623 regfile8: 00000003
624 regfile9: 00000000
625 regfile10: 00000000
626 regfile11: 00000000
627 regfile12: 00000000
628 regfile13: 00000000
629 regfile14: 00000000
630 regfile15: 00000000
631 regfile16: 00000000
632 regfile17: 00000000
633 regfile18: 00000000
634 regfile19: 00000000
635 regfile20: 00000000
636 regfile21: 00000000
637 regfile22: 00000000
638 regfile23: 00000000
639 regfile24: 00000000
640 regfile25: 00000000
641 regfile26: 00000000
642 regfile27: 00000000
643 regfile28: 00000000
644 regfile29: 00000000
645 regfile30: 00000000
646 regfile31: 00000000

```

```

16969 regfile0: 00000000
16970 regfile1: 00000040
16971 regfile2: 00000000
16972 regfile3: 00000000
16973 regfile4: 00000000
16974 regfile5: ffffffff
16975 regfile6: 00000001
16976 regfile7: 00000003
16977 regfile8: 00000003
16978 regfile9: 00000000
16979 regfile10: 00000000
16980 regfile11: 00000000
16981 regfile12: 00000000
16982 regfile13: 00000000
16983 regfile14: 00000000
16984 regfile15: 00000000
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 00000000
16988 regfile19: 00000000
16989 regfile20: 00000000
16990 regfile21: 00000000
16991 regfile22: 00000000
16992 regfile23: 00000000
16993 regfile24: 00000000
16994 regfile25: 00000000
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 00000000
16999 regfile30: 00000000
17000 regfile31: 00000000

```

+ F 632 lines unchanged, 16354 lines added, 14 lines modified, 0 lines deleted.

LWSW1:

TextDiff

File Edit Search Options Actions Help

D:\cpustest\my\my_16.26_lsws.txt D:\cpustest\my\my_16.26_lsws.txt

```

3301 regfile0: 00000000
3302 regfile1: 0000001f
3303 regfile2: 00000003
3304 regfile3: 00000004
3305 regfile4: 00000005
3306 regfile5: 00000006
3307 regfile6: 00000007
3308 regfile7: 00000008
3309 regfile8: 00000009
3310 regfile9: 0000000a
3311 regfile10: 0000000b
3312 regfile11: 0000000c
3313 regfile12: 0000000d
3314 regfile13: 0000000e
3315 regfile14: 0000000f
3316 regfile15: 00000010
3317 regfile16: 00000011
3318 regfile17: 00000012
3319 regfile18: 00000013
3320 regfile19: 00000014
3321 regfile20: 00000015
3322 regfile21: 00000016
3323 regfile22: 00000017
3324 regfile23: 00000018
3325 regfile24: 00000019
3326 regfile25: 0000001a
3327 regfile26: 0000001b
3328 regfile27: 0000001c
3329 regfile28: 0000001d
3330 regfile29: 0000001e
3331 regfile30: 00000000
3332 regfile31: 0000001f

```

```

16969 regfile0: 00000000
16970 regfile1: 0000001f
16971 regfile2: 00000003
16972 regfile3: 00000004
16973 regfile4: 00000005
16974 regfile5: 00000006
16975 regfile6: 00000007
16976 regfile7: 00000008
16977 regfile8: 00000009
16978 regfile9: 0000000a
16979 regfile10: 0000000b
16980 regfile11: 0000000c
16981 regfile12: 0000000d
16982 regfile13: 0000000e
16983 regfile14: 0000000f
16984 regfile15: 00000010
16985 regfile16: 00000011
16986 regfile17: 00000012
16987 regfile18: 00000013
16988 regfile19: 00000014
16989 regfile20: 00000015
16990 regfile21: 00000016
16991 regfile22: 00000017
16992 regfile23: 00000018
16993 regfile24: 00000019
16994 regfile25: 0000001a
16995 regfile26: 0000001b
16996 regfile27: 0000001c
16997 regfile28: 0000001d
16998 regfile29: 0000001e
16999 regfile30: 00000000
17000 regfile31: 0000001f

```

+ F 3177 lines unchanged, 13668 lines added, 155 lines modified, 0 lines deleted.

LWSW2:

XOR:

```
D:\cputest\my\my_16.26_lsw2.result.txt
3301 regfile0: 00000000
3302 regfile1: 0000000f
3303 regfile2: 000000ff
3304 regfile3: 00000fff
3305 regfile4: ffffffff
3306 regfile5: 00005555
3307 regfile6: fffffaaa
3308 regfile7: fffffbbb
3309 regfile8: fffffccc
3310 regfile9: fffffddd
3311 regfile10: fffffeee
3312 regfile11: ffffffff
3313 regfile12: fffff000
3314 regfile13: 000f0000
3315 regfile14: 00ff0000
3316 regfile15: 0fff0000
3317 regfile16: ffff0000
3318 regfile17: 55550000
3319 regfile18: aaaa0000
3320 regfile19: bbbb0000
3321 regfile20: cccc0000
3322 regfile21: dddd0000
3323 regfile22: eeee0000
3324 regfile23: ffff0000
3325 regfile24: fffff000
3326 regfile25: 000f0000
3327 regfile26: 00ff0000
3328 regfile27: 0fff0000
3329 regfile28: ffff0000
3330 regfile29: 55550000
3331 regfile30: 00000000
3332 regfile31: 0000000f

D:\cputest\my\my_16.26_lsw2.txt
16969 regfile0: 00000000
16970 regfile1: 0000000f
16971 regfile2: 000000ff
16972 regfile3: 00000fff
16973 regfile4: ffffffff
16974 regfile5: 00005555
16975 regfile6: fffffaaa
16976 regfile7: fffffbbb
16977 regfile8: fffffccc
16978 regfile9: fffffddd
16979 regfile10: fffffeee
16980 regfile11: ffffffff
16981 regfile12: fffff000
16982 regfile13: 000f0000
16983 regfile14: 00ff0000
16984 regfile15: 0fff0000
16985 regfile16: ffff0000
16986 regfile17: 55550000
16987 regfile18: aaaa0000
16988 regfile19: bbbb0000
16989 regfile20: cccc0000
16990 regfile21: dddd0000
16991 regfile22: eeee0000
16992 regfile23: ffff0000
16993 regfile24: fffff000
16994 regfile25: 000f0000
16995 regfile26: 00ff0000
16996 regfile27: 0fff0000
16997 regfile28: ffff0000
16998 regfile29: 55550000
16999 regfile30: 00000000
17000 regfile31: 0000000f
```

+ F 3177 lines unchanged, 13668 lines added, 155 lines modified, 0 lines deleted.

XOR:

```
D:\cputest\my\my_17_xor.result.txt
5103 regfile0: 00000000
5104 regfile1: 00000001
5105 regfile2: ffffffff
5106 regfile3: 80000000
5107 regfile4: 00005556
5108 regfile5: fffffaab
5109 regfile6: 80000003
5110 regfile7: 7ffffbbb9
5111 regfile8: 80007777
5112 regfile9: 7ffff999b
5113 regfile10: 80001111
5114 regfile11: 7ffffffd
5115 regfile12: 80003333
5116 regfile13: 80007774
5117 regfile14: 80002220
5118 regfile15: 7ffffccc
5119 regfile16: 80000002
5120 regfile17: 7ffffeee
5121 regfile18: 80006664
5122 regfile19: 5ddd5555
5123 regfile20: 19910000
5124 regfile21: f4650000
5125 regfile22: 7ffff8888
5126 regfile23: 7ffffffc
5127 regfile24: d20daaa8
5128 regfile25: 92335555
5129 regfile26: 3caaaaa8
5130 regfile27: e3545555
5131 regfile28: faadaaa8
5132 regfile29: a1215555
5133 regfile30: 99910000
5134 regfile31: 74650000

D:\cputest\my\my_17_xor.txt
16969 regfile0: 00000000
16970 regfile1: 00000001
16971 regfile2: ffffffff
16972 regfile3: 80000000
16973 regfile4: 00005556
16974 regfile5: fffffaab
16975 regfile6: 80000003
16976 regfile7: 7ffffbbb9
16977 regfile8: 80007777
16978 regfile9: 7ffff999b
16979 regfile10: 80001111
16980 regfile11: 7ffffffd
16981 regfile12: 80003333
16982 regfile13: 80007774
16983 regfile14: 80002220
16984 regfile15: 7ffffccc
16985 regfile16: 80000002
16986 regfile17: 7ffffeee
16987 regfile18: 80006664
16988 regfile19: 5ddd5555
16989 regfile20: 19910000
16990 regfile21: f4650000
16991 regfile22: 7ffff8888
16992 regfile23: 7ffffffc
16993 regfile24: d20daaa8
16994 regfile25: 92335555
16995 regfile26: 3caaaaa8
16996 regfile27: e3545555
16997 regfile28: faadaaa8
16998 regfile29: a1215555
16999 regfile30: 99910000
17000 regfile31: 74650000
```

+ F 4920 lines unchanged, 11866 lines added, 214 lines modified, 0 lines deleted.

NOR:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_18_nor.result.txt D:\cputest\my\my_18_nor.txt

```

5103 regfile0: 00000000
5104 regfile1: 00000001
5105 regfile2: ffffffff
5106 regfile3: 80000000
5107 regfile4: 00005556
5108 regfile5: fffffaab
5109 regfile6: 00005554
5110 regfile7: 7ffffaaa9
5111 regfile8: 00005555
5112 regfile9: 7ffffaaa9
5113 regfile10: 00005555
5114 regfile11: 7ffffaaa9
5115 regfile12: 00005555
5116 regfile13: 00005554
5117 regfile14: 7ffffaaa9
5118 regfile15: 00005555
5119 regfile16: 7ffffaaa9
5120 regfile17: 00005555
5121 regfile18: 7ffffaaa9
5122 regfile19: 00005555
5123 regfile20: 666effff
5124 regfile21: 0b9affff
5125 regfile22: 00005555
5126 regfile23: 00005554
5127 regfile24: 7ffffaaa9
5128 regfile25: 00005555
5129 regfile26: 7ffffaaa9
5130 regfile27: 00005555
5131 regfile28: 7ffffaaa9
5132 regfile29: 00005555
5133 regfile30: 99910000
5134 regfile31: 74650000

16969 regfile0: 00000000
16970 regfile1: 00000001
16971 regfile2: ffffffff
16972 regfile3: 80000000
16973 regfile4: 00005556
16974 regfile5: fffffaab
16975 regfile6: 00005554
16976 regfile7: 7ffffaaa9
16977 regfile8: 00005555
16978 regfile9: 7ffffaaa9
16979 regfile10: 00005555
16980 regfile11: 7ffffaaa9
16981 regfile12: 00005555
16982 regfile13: 00005554
16983 regfile14: 7ffffaaa9
16984 regfile15: 00005555
16985 regfile16: 7ffffaaa9
16986 regfile17: 00005555
16987 regfile18: 7ffffaaa9
16988 regfile19: 00005555
16989 regfile20: 666effff
16990 regfile21: 0b9affff
16991 regfile22: 00005555
16992 regfile23: 00005554
16993 regfile24: 7ffffaaa9
16994 regfile25: 00005555
16995 regfile26: 7ffffaaa9
16996 regfile27: 00005555
16997 regfile28: 7ffffaaa9
16998 regfile29: 00005555
16999 regfile30: 99910000
17000 regfile31: 74650000

```

+ F 4920 lines unchanged, 11866 lines added, 214 lines modified, 0 lines deleted.

OR:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_19_or.result.txt D:\cputest\my\my_19_or.txt

```

5103 regfile0: 00000000
5104 regfile1: 00000001
5105 regfile2: ffffffff
5106 regfile3: 80000000
5107 regfile4: 00005556
5108 regfile5: fffffaab
5109 regfile6: ffffffff
5110 regfile7: ffffffff
5111 regfile8: ffffffff
5112 regfile9: ffffffff
5113 regfile10: ffffffff
5114 regfile11: ffffffff
5115 regfile12: ffffffff
5116 regfile13: ffffffff
5117 regfile14: ffffffff
5118 regfile15: ffffffff
5119 regfile16: ffffffff
5120 regfile17: ffffffff
5121 regfile18: ffffffff
5122 regfile19: ffffffff
5123 regfile20: 99910000
5124 regfile21: f4650000
5125 regfile22: ffffffff
5126 regfile23: ffffffff
5127 regfile24: ffffffff
5128 regfile25: ffffffff
5129 regfile26: ffffffff
5130 regfile27: ffffffff
5131 regfile28: ffffffff
5132 regfile29: ffffffff
5133 regfile30: 99910000
5134 regfile31: 74650000

16969 regfile0: 00000000
16970 regfile1: 00000001
16971 regfile2: ffffffff
16972 regfile3: 80000000
16973 regfile4: 00005556
16974 regfile5: fffffaab
16975 regfile6: ffffffff
16976 regfile7: ffffffff
16977 regfile8: ffffffff
16978 regfile9: ffffffff
16979 regfile10: ffffffff
16980 regfile11: ffffffff
16981 regfile12: ffffffff
16982 regfile13: ffffffff
16983 regfile14: ffffffff
16984 regfile15: ffffffff
16985 regfile16: ffffffff
16986 regfile17: ffffffff
16987 regfile18: ffffffff
16988 regfile19: ffffffff
16989 regfile20: 99910000
16990 regfile21: f4650000
16991 regfile22: ffffffff
16992 regfile23: ffffffff
16993 regfile24: ffffffff
16994 regfile25: ffffffff
16995 regfile26: ffffffff
16996 regfile27: ffffffff
16997 regfile28: ffffffff
16998 regfile29: ffffffff
16999 regfile30: 99910000
17000 regfile31: 74650000

```

+ F 5023 lines unchanged, 11866 lines added, 111 lines modified, 0 lines deleted.

SLL:

TextDiff

File Edit Search Options Actions Help

D:\cpustest20_sll.result.txt D:\cpustest\my\my_20_sll.txt

```

1091 regfile0: 00000000
1092 regfile1: 0000000f
1093 regfile2: 00005aff
1094 regfile3: ffffff000
1095 regfile4: ffffff000
1096 regfile5: ffffff5a
1097 regfile6: 0000001e
1098 regfile7: 80000000
1099 regfile8: 0000b5fe
1100 regfile9: 80000000
1101 regfile10: fffffe000
1102 regfile11: 00000000
1103 regfile12: fffffe000
1104 regfile13: 00000000
1105 regfile14: ffffffeb4
1106 regfile15: 00000000
1107 regfile16: 0000001e0
1108 regfile17: 000f0000
1109 regfile18: 00016bfc
1110 regfile19: d7f80000
1111 regfile20: fff80000
1112 regfile21: ffe00000
1113 regfile22: ffff80000
1114 regfile23: ffcc0000
1115 regfile24: 68000000
1116 regfile25: 80000000
1117 regfile26: 00000000
1118 regfile27: 00000000
1119 regfile28: 00000000
1120 regfile29: fffffffe
1121 regfile30: 80000000
1122 regfile31: ffffffff

```

```

16969 regfile0: 00000000
16970 regfile1: 0000000f
16971 regfile2: 00005aff
16972 regfile3: ffffff000
16973 regfile4: ffffff000
16974 regfile5: ffffff5a
16975 regfile6: 0000001e
16976 regfile7: 80000000
16977 regfile8: 0000b5fe
16978 regfile9: 80000000
16979 regfile10: fffffe000
16980 regfile11: 00000000
16981 regfile12: fffffe000
16982 regfile13: 00000000
16983 regfile14: ffffffeb4
16984 regfile15: 00000000
16985 regfile16: 0000001e0
16986 regfile17: 000f0000
16987 regfile18: 00016bfc
16988 regfile19: d7f80000
16989 regfile20: fff80000
16990 regfile21: ffe00000
16991 regfile22: ffff80000
16992 regfile23: ffcc0000
16993 regfile24: 68000000
16994 regfile25: 80000000
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: fffffffe
16999 regfile30: 80000000
17000 regfile31: ffffffff

```

+ F 1061 lines unchanged, 15878 lines added, 61 lines modified, 0 lines deleted.

SLLV:

TextDiff

File Edit Search Options Actions Help

D:\cpustest21_sllv.result.txt D:\cpustest\my\my_21_sllv.txt

```

1941 regfile0: 00000000
1942 regfile1: 0000000f
1943 regfile2: 00005aff
1944 regfile3: ffffff000
1945 regfile4: f0000000
1946 regfile5: ffffffff
1947 regfile6: 0000001e
1948 regfile7: 80000000
1949 regfile8: 0000b5fe
1950 regfile9: 80000000
1951 regfile10: fffffe000
1952 regfile11: 00000000
1953 regfile12: e0000000
1954 regfile13: 00000000
1955 regfile14: fffffffe
1956 regfile15: 80000000
1957 regfile16: 0000001e0
1958 regfile17: 000f0000
1959 regfile18: 00016bfc
1960 regfile19: d7f80000
1961 regfile20: fff80000
1962 regfile21: ffe00000
1963 regfile22: 80000000
1964 regfile23: 00000000
1965 regfile24: fc000000
1966 regfile25: c0000000
1967 regfile26: 00000000
1968 regfile27: 00000000
1969 regfile28: 00000000
1970 regfile29: 00000002
1971 regfile30: 80000000
1972 regfile31: 0000001f

```

```

16969 regfile0: 00000000
16970 regfile1: 0000000f
16971 regfile2: 00005aff
16972 regfile3: ffffff000
16973 regfile4: f0000000
16974 regfile5: ffffffff
16975 regfile6: 0000001e
16976 regfile7: 80000000
16977 regfile8: 0000b5fe
16978 regfile9: 80000000
16979 regfile10: fffffe000
16980 regfile11: 00000000
16981 regfile12: e0000000
16982 regfile13: 00000000
16983 regfile14: fffffffe
16984 regfile15: 80000000
16985 regfile16: 0000001e0
16986 regfile17: 000f0000
16987 regfile18: 00016bfc
16988 regfile19: d7f80000
16989 regfile20: fff80000
16990 regfile21: ffe00000
16991 regfile22: 80000000
16992 regfile23: 00000000
16993 regfile24: fc000000
16994 regfile25: c0000000
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 00000002
16999 regfile30: 80000000
17000 regfile31: 0000001f

```

+ F 1872 lines unchanged, 15028 lines added, 100 lines modified, 0 lines deleted.

SLTU:

TextDiff

File Edit Search Options Actions Help

D:\cputest\22_sltu.result.txt D:\cputest\my\my_22_sltu.txt

```

2213 regfile0: 00000000
2214 regfile1: 00000001
2215 regfile2: 00000001
2216 regfile3: 00000001
2217 regfile4: 00000001
2218 regfile5: 00000001
2219 regfile6: 00000000
2220 regfile7: 00000001
2221 regfile8: 00000000
2222 regfile9: 00000000
2223 regfile10: 00000000
2224 regfile11: 00000000
2225 regfile12: 00000001
2226 regfile13: 00000000
2227 regfile14: 00000001
2228 regfile15: 00000001
2229 regfile16: 00000000
2230 regfile17: 00000001
2231 regfile18: 00000000
2232 regfile19: 00000001
2233 regfile20: 00000001
2234 regfile21: 00000001
2235 regfile22: 00000001
2236 regfile23: 00000000
2237 regfile24: 00000000
2238 regfile25: 00000001
2239 regfile26: 00000000
2240 regfile27: 00000001
2241 regfile28: 00000000
2242 regfile29: 00000001
2243 regfile30: 00000001
2244 regfile31: 00000000

16969 regfile0: 00000000
16970 regfile1: 00000001
16971 regfile2: 00000001
16972 regfile3: 00000001
16973 regfile4: 00000001
16974 regfile5: 00000001
16975 regfile6: 00000000
16976 regfile7: 00000001
16977 regfile8: 00000000
16978 regfile9: 00000000
16979 regfile10: 00000000
16980 regfile11: 00000000
16981 regfile12: 00000001
16982 regfile13: 00000000
16983 regfile14: 00000001
16984 regfile15: 00000001
16985 regfile16: 00000000
16986 regfile17: 00000001
16987 regfile18: 00000000
16988 regfile19: 00000001
16989 regfile20: 00000001
16990 regfile21: 00000001
16991 regfile22: 00000001
16992 regfile23: 00000000
16993 regfile24: 00000000
16994 regfile25: 00000001
16995 regfile26: 00000000
16996 regfile27: 00000001
16997 regfile28: 00000000
16998 regfile29: 00000001
16999 regfile30: 00000001
17000 regfile31: 00000000

```

+ F 2119 lines unchanged, 14756 lines added, 125 lines modified, 0 lines deleted.

SRA:

TextDiff

File Edit Search Options Actions Help

D:\cputest\23_sra.result.txt D:\cputest\my\my_23_sra.txt

```

1091 regfile0: 00000000
1092 regfile1: 0000000f
1093 regfile2: 00005af1
1094 regfile3: ffffff000
1095 regfile4: ffffff000
1096 regfile5: ffffff5a
1097 regfile6: 00000007
1098 regfile7: 00000000
1099 regfile8: 00002d7f
1100 regfile9: 00000000
1101 regfile10: ffffff800
1102 regfile11: ffffffff
1103 regfile12: ffffff800
1104 regfile13: ffffffff
1105 regfile14: ffffffad
1106 regfile15: ffffffff
1107 regfile16: 00000000
1108 regfile17: 00000000
1109 regfile18: 000016bf
1110 regfile19: 00000000
1111 regfile20: ffffffe0
1112 regfile21: ffffff8
1113 regfile22: ffffffe00
1114 regfile23: ffffffc0
1115 regfile24: ffffffff
1116 regfile25: ffffffff
1117 regfile26: 00000000
1118 regfile27: 00000000
1119 regfile28: 00000000
1120 regfile29: ffffffff
1121 regfile30: ffffffff
1122 regfile31: ffffffff

16969 regfile0: 00000000
16970 regfile1: 0000000f
16971 regfile2: 00005af1
16972 regfile3: ffffff000
16973 regfile4: ffffff000
16974 regfile5: ffffff5a
16975 regfile6: 00000007
16976 regfile7: 00000000
16977 regfile8: 00002d7f
16978 regfile9: 00000000
16979 regfile10: ffffff800
16980 regfile11: ffffffff
16981 regfile12: ffffff800
16982 regfile13: ffffffff
16983 regfile14: ffffffad
16984 regfile15: ffffffff
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 000016bf
16988 regfile19: 00000000
16989 regfile20: ffffffe0
16990 regfile21: ffffff8
16991 regfile22: ffffffe00
16992 regfile23: ffffffc0
16993 regfile24: ffffffff
16994 regfile25: ffffffff
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: ffffffff
16999 regfile30: ffffffff
17000 regfile31: ffffffff

```

+ F 1061 lines unchanged, 15878 lines added, 61 lines modified, 0 lines deleted.

SRL:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_24_srl.txt D:\cputest\my\my_24_sr.txt

```

1091 regfile0: 00000000
1092 regfile1: 0000000f
1093 regfile2: 00005aff
1094 regfile3: ffffff000
1095 regfile4: ffffff000
1096 regfile5: ffffff5a
1097 regfile6: 00000007
1098 regfile7: 00000000
1099 regfile8: 00002d7f
1100 regfile9: 00000000
1101 regfile10: 7ffff800
1102 regfile11: 00000001
1103 regfile12: 7ffff800
1104 regfile13: 00000001
1105 regfile14: 7fffffad
1106 regfile15: 00000001
1107 regfile16: 00000000
1108 regfile17: 00000000
1109 regfile18: 000016bf
1110 regfile19: 00000000
1111 regfile20: 01ffffe0
1112 regfile21: 007ffff8
1113 regfile22: 1fffffe00
1114 regfile23: 03ffffc0
1115 regfile24: 0000003f
1116 regfile25: 00000003
1117 regfile26: 00000000
1118 regfile27: 00000000
1119 regfile28: 00000000
1120 regfile29: 7fffffff
1121 regfile30: 00000001
1122 regfile31: ffffffff

```

```

16969 regfile0: 00000000
16970 regfile1: 0000000f
16971 regfile2: 00005aff
16972 regfile3: ffffff000
16973 regfile4: ffffff000
16974 regfile5: ffffff5a
16975 regfile6: 00000007
16976 regfile7: 00000000
16977 regfile8: 00002d7f
16978 regfile9: 00000000
16979 regfile10: 7ffff800
16980 regfile11: 00000001
16981 regfile12: 7ffff800
16982 regfile13: 00000001
16983 regfile14: 7fffffad
16984 regfile15: 00000001
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 000016bf
16988 regfile19: 00000000
16989 regfile20: 01ffffe0
16990 regfile21: 007ffff8
16991 regfile22: 1fffffe00
16992 regfile23: 03ffffc0
16993 regfile24: 0000003f
16994 regfile25: 00000003
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 7fffffff
16999 regfile30: 00000001
17000 regfile31: ffffffff

```

+ F 1061 lines unchanged, 15878 lines added, 61 lines modified, 0 lines deleted.

SUBU:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_25_subu.txt D:\cputest\my\my_25_sub.txt

```

5035 regfile0: 00000000
5036 regfile1: 00000001
5037 regfile2: ffffffff
5038 regfile3: 80000000
5039 regfile4: 80000000
5040 regfile5: 80000001
5041 regfile6: 80000002
5042 regfile7: 80000004
5043 regfile8: 80000004
5044 regfile9: 80000005
5045 regfile10: 80000006
5046 regfile11: 80000007
5047 regfile12: 80000008
5048 regfile13: 80000009
5049 regfile14: 8000000a
5050 regfile15: 8000000b
5051 regfile16: 8000000c
5052 regfile17: 8000000d
5053 regfile18: 8000000e
5054 regfile19: 70000000
5055 regfile20: 7fff8000
5056 regfile21: 00000000
5057 regfile22: 7fffffff
5058 regfile23: 7fffff00
5059 regfile24: 0fff0000
5060 regfile25: 1fff0000
5061 regfile26: 6fff0000
5062 regfile27: 0f300000
5063 regfile28: 001c0000
5064 regfile29: 001d0000
5065 regfile30: 001e0000
5066 regfile31: 001f0000

```

```

16969 regfile0: 00000000
16970 regfile1: 00000001
16971 regfile2: ffffffff
16972 regfile3: 80000000
16973 regfile4: 80000000
16974 regfile5: 80000001
16975 regfile6: 80000002
16976 regfile7: 80000004
16977 regfile8: 80000004
16978 regfile9: 80000005
16979 regfile10: 80000006
16980 regfile11: 80000007
16981 regfile12: 80000008
16982 regfile13: 80000009
16983 regfile14: 8000000a
16984 regfile15: 8000000b
16985 regfile16: 8000000c
16986 regfile17: 8000000d
16987 regfile18: 8000000e
16988 regfile19: 70000000
16989 regfile20: 7fff8000
16990 regfile21: 00000000
16991 regfile22: 7fffffff
16992 regfile23: 7fffff00
16993 regfile24: 0fff0000
16994 regfile25: 1fff0000
16995 regfile26: 6fff0000
16996 regfile27: 0f300000
16997 regfile28: 001c0000
16998 regfile29: 001d0000
16999 regfile30: 001e0000
17000 regfile31: 001f0000

```

+ F 4910 lines unchanged, 11934 lines added, 156 lines modified, 0 lines deleted.

ADD: (当时给的测试文件是错的，因此只能执行到 pc=0000018c)

SUB:

```
D:\cputest\my\my_27_add.result.txt
```

```
D:\cputest\my\my_27_add.txt
```

```
+ F 3282 lines unchanged, 13598 lines added, 120 lines modified, 0 lines deleted.
```

SLT:

```
D:\cputest\my\my_28_sub.result.txt
```

```
D:\cputest\my\my_28_sub.txt
```

```
+ F 3919 lines unchanged, 12954 lines added, 127 lines modified, 0 lines deleted.
```

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_29_slt.txt

```

2213 regfile0: 00000000
2214 regfile1: 00000001
2215 regfile2: 00000001
2216 regfile3: 00000001
2217 regfile4: 00000001
2218 regfile5: 00000001
2219 regfile6: 00000000
2220 regfile7: 00000001
2221 regfile8: 00000001
2222 regfile9: 00000001
2223 regfile10: 00000000
2224 regfile11: 00000001
2225 regfile12: 00000000
2226 regfile13: 00000000
2227 regfile14: 00000001
2228 regfile15: 00000000
2229 regfile16: 00000000
2230 regfile17: 00000000
2231 regfile18: 00000000
2232 regfile19: 00000001
2233 regfile20: 00000000
2234 regfile21: 00000001
2235 regfile22: 00000000
2236 regfile23: 00000000
2237 regfile24: 00000000
2238 regfile25: 00000000
2239 regfile26: 00000000
2240 regfile27: 00000000
2241 regfile28: 00000000
2242 regfile29: 00000001
2243 regfile30: 00000000
2244 regfile31: 00000000

```

D:\cputest\my\my_29_slt.txt

```

16969 regfile0: 00000000
16970 regfile1: 00000001
16971 regfile2: 00000001
16972 regfile3: 00000001
16973 regfile4: 00000001
16974 regfile5: 00000001
16975 regfile6: 00000000
16976 regfile7: 00000001
16977 regfile8: 00000001
16978 regfile9: 00000001
16979 regfile10: 00000000
16980 regfile11: 00000001
16981 regfile12: 00000000
16982 regfile13: 00000000
16983 regfile14: 00000001
16984 regfile15: 00000000
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 00000000
16988 regfile19: 00000001
16989 regfile20: 00000000
16990 regfile21: 00000001
16991 regfile22: 00000000
16992 regfile23: 00000000
16993 regfile24: 00000000
16994 regfile25: 00000000
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 00000001
16999 regfile30: 00000000
17000 regfile31: 00000000

```

+ F 2119 lines unchanged, 14756 lines added, 125 lines modified, 0 lines deleted.

SRLV:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_30_srsv.txt

```

1941 regfile0: 00000000
1942 regfile1: 0000000f
1943 regfile2: 00005aff
1944 regfile3: ffffff00
1945 regfile4: f0000000
1946 regfile5: ffffffff
1947 regfile6: 00000007
1948 regfile7: 00000000
1949 regfile8: 00002d7f
1950 regfile9: 00000000
1951 regfile10: 7ffff800
1952 regfile11: 00000001
1953 regfile12: 78000000
1954 regfile13: 00000001
1955 regfile14: 7fffffff
1956 regfile15: 00000001
1957 regfile16: 00000000
1958 regfile17: 00000000
1959 regfile18: 000016bf
1960 regfile19: 00000000
1961 regfile20: 01ffffe0
1962 regfile21: 007ffff8
1963 regfile22: 1e000000
1964 regfile23: 03c00000
1965 regfile24: 0000003f
1966 regfile25: 00000003
1967 regfile26: 00000000
1968 regfile27: 00000000
1969 regfile28: 00000000
1970 regfile29: 00000000
1971 regfile30: 00000000
1972 regfile31: 0000001f

```

D:\cputest\my\my_30_srsv.txt

```

16969 regfile0: 00000000
16970 regfile1: 0000000f
16971 regfile2: 00005aff
16972 regfile3: ffffff00
16973 regfile4: f0000000
16974 regfile5: ffffffff
16975 regfile6: 00000007
16976 regfile7: 00000000
16977 regfile8: 00002d7f
16978 regfile9: 00000000
16979 regfile10: 7ffff800
16980 regfile11: 00000001
16981 regfile12: 78000000
16982 regfile13: 00000001
16983 regfile14: 7fffffff
16984 regfile15: 00000001
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 000016bf
16988 regfile19: 00000000
16989 regfile20: 01ffffe0
16990 regfile21: 007ffff8
16991 regfile22: 1e000000
16992 regfile23: 03c00000
16993 regfile24: 0000003f
16994 regfile25: 00000003
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 00000000
16999 regfile30: 00000000
17000 regfile31: 0000001f

```

+ F 1879 lines unchanged, 15028 lines added, 93 lines modified, 0 lines deleted.

SRAV:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_31_srav.result.txt D:\cputest\my\my_31_srav.txt

```

1941 regfile0: 00000000
1942 regfile1: 0000000f
1943 regfile2: 00005aff
1944 regfile3: ffffff00
1945 regfile4: f0000000
1946 regfiles: ffffffff
1947 regfile6: 00000007
1948 regfile7: 00000000
1949 regfile8: 00002d7f
1950 regfile9: 00000000
1951 regfile10: fffff800
1952 regfile11: ffffffff
1953 regfile12: f8000000
1954 regfile13: ffffffff
1955 regfile14: ffffffff
1956 regfile15: ffffffff
1957 regfile16: 00000000
1958 regfile17: 00000000
1959 regfile18: 000016bf
1960 regfile19: 00000000
1961 regfile20: ffffffe0
1962 regfile21: ffffff88
1963 regfile22: fe000000
1964 regfile23: ffco0000
1965 regfile24: ffffffff
1966 regfile25: ffffffff
1967 regfile26: 00000000
1968 regfile27: 00000000
1969 regfile28: 00000000
1970 regfile29: 00000000
1971 regfile30: 00000000
1972 regfile31: 0000001f

16969 regfile0: 00000000
16970 regfile1: 0000000f
16971 regfile2: 00005aff
16972 regfile3: ffffff00
16973 regfile4: f0000000
16974 regfile5: ffffffff
16975 regfile6: 00000007
16976 regfile7: 00000000
16977 regfile8: 00002d7f
16978 regfile9: 00000000
16979 regfile10: fffff800
16980 regfile11: ffffffff
16981 regfile12: f8000000
16982 regfile13: ffffffff
16983 regfile14: ffffffff
16984 regfile15: ffffffff
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 000016bf
16988 regfile19: 00000000
16989 regfile20: ffffffe0
16990 regfile21: ffffff88
16991 regfile22: fe000000
16992 regfile23: ffco0000
16993 regfile24: ffffffff
16994 regfile25: ffffffff
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 00000000
16999 regfile30: 00000000
17000 regfile31: 0000001f

```

+ ~ - 1879 lines unchanged, 15028 lines added, 93 lines modified, 0 lines deleted.

CLZ:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_32_clz.result.txt D:\cputest\my\my_32_clz.txt

```

2247 regfile0: 00000000
2248 regfile1: 00000000
2249 regfile2: 0000001e
2250 regfile3: 0000001d
2251 regfile4: 0000001c
2252 regfile5: 0000001b
2253 regfile6: 0000001a
2254 regfile7: 00000019
2255 regfile8: 00000018
2256 regfile9: 00000017
2257 regfile10: 00000016
2258 regfile11: 00000015
2259 regfile12: 00000014
2260 regfile13: 00000013
2261 regfile14: 00000012
2262 regfile15: 00000011
2263 regfile16: 00000010
2264 regfile17: 0000000f
2265 regfile18: 0000000e
2266 regfile19: 0000000d
2267 regfile20: 0000000c
2268 regfile21: 0000000b
2269 regfile22: 0000000a
2270 regfile23: 00000009
2271 regfile24: 00000008
2272 regfile25: 00000007
2273 regfile26: 00000006
2274 regfile27: 00000005
2275 regfile28: 00000004
2276 regfile29: 00000003
2277 regfile30: 00000002
2278 regfile31: 00000001

16969 regfile0: 00000000
16970 regfile1: 00000000
16971 regfile2: 0000001e
16972 regfile3: 0000001d
16973 regfile4: 0000001c
16974 regfile5: 0000001b
16975 regfile6: 0000001a
16976 regfile7: 00000019
16977 regfile8: 00000018
16978 regfile9: 00000017
16979 regfile10: 00000016
16980 regfile11: 00000015
16981 regfile12: 00000014
16982 regfile13: 00000013
16983 regfile14: 00000012
16984 regfile15: 00000011
16985 regfile16: 00000010
16986 regfile17: 0000000f
16987 regfile18: 0000000e
16988 regfile19: 0000000d
16989 regfile20: 0000000c
16990 regfile21: 0000000b
16991 regfile22: 0000000a
16992 regfile23: 00000009
16993 regfile24: 00000008
16994 regfile25: 00000007
16995 regfile26: 00000006
16996 regfile27: 00000005
16997 regfile28: 00000004
16998 regfile29: 00000003
16999 regfile30: 00000002
17000 regfile31: 00000001

```

DIVU:

JALU:

```
D:\cputest\33_divu.result.txt
```

```
D:\cputest\my\my_33_divu.txt
```

```

14691 regfile0: 00000000
14692 regfile1: ffffffa
14693 regfile2: 00000000
14694 regfile3: ffffffa
14695 regfile4: ffffffd
14696 regfile5: ffffffa
14697 regfile6: ffffffd
14698 regfile7: ffffffa
14699 regfile8: ffffffd
14700 regfile9: ffffffa
14701 regfile10: ffffffd
14702 regfile11: ffffffa
14703 regfile12: ffffffd
14704 regfile13: ffffffa
14705 regfile14: ffffffd
14706 regfile15: ffffffa
14707 regfile16: ffffffd
14708 regfile17: ffffffa
14709 regfile18: ffffffd
14710 regfile19: ffffffa
14711 regfile20: ffffffd
14712 regfile21: ffffffa
14713 regfile22: ffffffd
14714 regfile23: ffffffa
14715 regfile24: ffffffd
14716 regfile25: ffffffa
14717 regfile26: ffffffd
14718 regfile27: ffffffa
14719 regfile28: ffffffd
14720 regfile29: ffffffa
14721 regfile30: ffffffa
14722 regfile31: ffffffd

40769 regfile0: 00000000
40770 regfile1: ffffffa
40771 regfile2: 00000000
40772 regfile3: ffffffa
40773 regfile4: ffffffd
40774 regfile5: ffffffa
40775 regfile6: ffffffd
40776 regfile7: ffffffa
40777 regfile8: ffffffd
40778 regfile9: ffffffa
40779 regfile10: ffffffd
40780 regfile11: ffffffa
40781 regfile12: ffffffd
40782 regfile13: ffffffa
40783 regfile14: ffffffd
40784 regfile15: ffffffa
40785 regfile16: ffffffd
40786 regfile17: ffffffa
40787 regfile18: ffffffd
40788 regfile19: ffffffa
40789 regfile20: ffffffd
40790 regfile21: ffffffa
40791 regfile22: ffffffd
40792 regfile23: ffffffa
40793 regfile24: ffffffd
40794 regfile25: ffffffa
40795 regfile26: ffffffd
40796 regfile27: ffffffa
40797 regfile28: ffffffd
40798 regfile29: ffffffa
40799 regfile30: ffffffa
40800 regfile31: ffffffd

```

+ Fil 14244 lines unchanged, 26078 lines added, 478 lines modified, 0 lines deleted.

JALR:

```
D:\cputest\35_jalr.result.txt
```

```
D:\cputest\my\my_35_jalr.txt
```

```

275 regfile0: 00000000
276 regfile1: 00400018
277 regfile2: 00000000
278 regfile3: 00000000
279 regfile4: 00000001
280 regfile5: 00000001
281 regfile6: 00000002
282 regfile7: 00000000
283 regfile8: 00000000
284 regfile9: 00000000
285 regfile10: 00000000
286 regfile11: 00000000
287 regfile12: 00000000
288 regfile13: 00000000
289 regfile14: 00000000
290 regfile15: 00000000
291 regfile16: 00000000
292 regfile17: 00000000
293 regfile18: 00000000
294 regfile19: 00000000
295 regfile20: 00000000
296 regfile21: 00000000
297 regfile22: 00000000
298 regfile23: 00000000
299 regfile24: 00000000
300 regfile25: 00000000
301 regfile26: 00000000
302 regfile27: 00000000
303 regfile28: 00000000
304 regfile29: 00000000
305 regfile30: 00400010
306 regfile31: 00000000

16969 regfile0: 00000000
16970 regfile1: 00400018
16971 regfile2: 00000000
16972 regfile3: 00000000
16973 regfile4: 00000001
16974 regfile5: 00000001
16975 regfile6: 00000002
16976 regfile7: 00000000
16977 regfile8: 00000000
16978 regfile9: 00000000
16979 regfile10: 00000000
16980 regfile11: 00000000
16981 regfile12: 00000000
16982 regfile13: 00000000
16983 regfile14: 00000000
16984 regfile15: 00000000
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 00000000
16988 regfile19: 00000000
16989 regfile20: 00000000
16990 regfile21: 00000000
16991 regfile22: 00000000
16992 regfile23: 00000000
16993 regfile24: 00000000
16994 regfile25: 00000000
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 00000000
16999 regfile30: 00400010
17000 regfile31: 00000000

```

+ Fil 292 lines unchanged, 16694 lines added, 14 lines modified, 0 lines deleted.

LBSB:

LBSB:

```
D:\cputest\my\my_36.39_lsb.txt
3301 regfile0: 00000000
3302 regfile1: 0000001f
3303 regfile2: 00000003
3304 regfile3: 00000004
3305 regfile4: 00000005
3306 regfile5: 00000006
3307 regfile6: 00000007
3308 regfile7: 00000008
3309 regfile8: 00000009
3310 regfile9: 0000000a
3311 regfile10: 0000000b
3312 regfile11: 0000000c
3313 regfile12: 0000000d
3314 regfile13: 0000000e
3315 regfile14: 0000000f
3316 regfile15: 00000010
3317 regfile16: 00000011
3318 regfile17: 00000012
3319 regfile18: 00000013
3320 regfile19: 00000014
3321 regfile20: 00000015
3322 regfile21: 00000016
3323 regfile22: 00000017
3324 regfile23: 00000018
3325 regfile24: 00000019
3326 regfile25: 0000001a
3327 regfile26: 0000001b
3328 regfile27: 0000001c
3329 regfile28: 0000001d
3330 regfile29: 0000001e
3331 regfile30: 00000000
3332 regfile31: 0000001f

D:\cputest\my\my_36.39_lsb.result.txt
16969 regfile0: 00000000
16970 regfile1: 00000000
16971 regfile2: 00000003
16972 regfile3: 00000004
16973 regfile4: 00000005
16974 regfile5: 00000006
16975 regfile6: 00000007
16976 regfile7: 00000008
16977 regfile8: 00000009
16978 regfile9: 0000000a
16979 regfile10: 0000000b
16980 regfile11: 0000000c
16981 regfile12: 0000000d
16982 regfile13: 0000000e
16983 regfile14: 0000000f
16984 regfile15: 00000010
16985 regfile16: 00000011
16986 regfile17: 00000012
16987 regfile18: 00000013
16988 regfile19: 00000014
16989 regfile20: 00000015
16990 regfile21: 00000016
16991 regfile22: 00000017
16992 regfile23: 00000018
16993 regfile24: 00000019
16994 regfile25: 0000001a
16995 regfile26: 0000001b
16996 regfile27: 0000001c
16997 regfile28: 0000001d
16998 regfile29: 0000001e
16999 regfile30: 00000000
17000 regfile31: 0000001f
```

+ Fil 2716 lines unchanged, 13668 lines added, 616 lines modified, 0 lines deleted.

LBSB2:

```
D:\cputest\my\my_36.39_lsb2.txt
3301 regfile0: 00000000
3302 regfile1: 00000000
3303 regfile2: ffffffff
3304 regfile3: ffffffff
3305 regfile4: ffffffff
3306 regfile5: 00000055
3307 regfile6: ffffffaa
3308 regfile7: ffffffb
3309 regfile8: ffffffc
3310 regfile9: ffffffd
3311 regfile10: ffffffe
3312 regfile11: fffffff
3313 regfile12: 00000000
3314 regfile13: 00000000
3315 regfile14: 00000000
3316 regfile15: 00000000
3317 regfile16: 00000000
3318 regfile17: 00000000
3319 regfile18: 00000000
3320 regfile19: 00000000
3321 regfile20: 00000000
3322 regfile21: 00000000
3323 regfile22: 00000000
3324 regfile23: 00000000
3325 regfile24: 00000000
3326 regfile25: 00000000
3327 regfile26: 00000000
3328 regfile27: 00000000
3329 regfile28: 00000000
3330 regfile29: 00000000
3331 regfile30: 00000000
3332 regfile31: 000f0000

D:\cputest\my\my_36.39_lsb2.result.txt
16969 regfile0: 00000000
16970 regfile1: 00000000
16971 regfile2: ffffffff
16972 regfile3: ffffffff
16973 regfile4: ffffffff
16974 regfile5: 00000055
16975 regfile6: ffffffaa
16976 regfile7: ffffffb
16977 regfile8: ffffffc
16978 regfile9: ffffffd
16979 regfile10: ffffffe
16980 regfile11: fffffff
16981 regfile12: 00000000
16982 regfile13: 00000000
16983 regfile14: 00000000
16984 regfile15: 00000000
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 00000000
16988 regfile19: 00000000
16989 regfile20: 00000000
16990 regfile21: 00000000
16991 regfile22: 00000000
16992 regfile23: 00000000
16993 regfile24: 00000000
16994 regfile25: 00000000
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 00000000
16999 regfile30: 00000000
17000 regfile31: 000f0000
```

+ Fil 2666 lines unchanged, 13668 lines added, 646 lines modified, 0 lines deleted.

LBU:

TextDiff

File Edit Search Options Actions Help

```
D:\cputest\37_lbu.result.txt          D:\cputest\my\my_37_lbu.txt
3301 regfile0: 00000000              16969 regfile0: 00000000
3302 regfile1: 0000001f              16970 regfile1: 00000000
3303 regfile2: 00000003              16971 regfile2: 00000003
3304 regfile3: 00000004              16972 regfile3: 00000004
3305 regfile4: 00000005              16973 regfile4: 00000005
3306 regfile5: 00000006              16974 regfile5: 00000006
3307 regfile6: 00000007              16975 regfile6: 00000007
3308 regfile7: 00000008              16976 regfile7: 00000008
3309 regfile8: 00000009              16977 regfile8: 00000009
3310 regfile9: 0000000a              16978 regfile9: 0000000a
3311 regfile10: 0000000b             16979 regfile10: 0000000b
3312 regfile11: 0000000c             16980 regfile11: 0000000c
3313 regfile12: 0000000d             16981 regfile12: 0000000d
3314 regfile13: 0000000e             16982 regfile13: 0000000e
3315 regfile14: 0000000f             16983 regfile14: 0000000f
3316 regfile15: 00000010             16984 regfile15: 00000010
3317 regfile16: 00000011             16985 regfile16: 00000011
3318 regfile17: 00000012             16986 regfile17: 00000012
3319 regfile18: 00000013             16987 regfile18: 00000013
3320 regfile19: 00000014             16988 regfile19: 00000014
3321 regfile20: 00000015             16989 regfile20: 00000015
3322 regfile21: 00000016             16990 regfile21: 00000016
3323 regfile22: 00000017             16991 regfile22: 00000017
3324 regfile23: 00000018             16992 regfile23: 00000018
3325 regfile24: 00000019             16993 regfile24: 00000019
3326 regfile25: 0000001a             16994 regfile25: 0000001a
3327 regfile26: 0000001b             16995 regfile26: 0000001b
3328 regfile27: 0000001c             16996 regfile27: 0000001c
3329 regfile28: 0000001d             16997 regfile28: 0000001d
3330 regfile29: 0000001e             16998 regfile29: 0000001e
3331 regfile30: 00000000             16999 regfile30: 00000000
3332 regfile31: 0000001f             17000 regfile31: 0000001f

+ File 2684 lines unchanged, 13668 lines added, 648 lines modified, 0 lines deleted.
```

LBU2:

TextDiff

File Edit Search Options Actions Help

```
D:\cputest\37_lbu2.result.txt          D:\cputest\my\my_37_lbu2.txt
3301 regfile0: 00000000              16969 regfile0: 00000000
3302 regfile1: 00000000              16970 regfile1: 00000000
3303 regfile2: 000000ff              16971 regfile2: 000000ff
3304 regfile3: 000000ff              16972 regfile3: 000000ff
3305 regfile4: 000000ff              16973 regfile4: 000000ff
3306 regfile5: 00000055              16974 regfile5: 00000055
3307 regfile6: 000000aa              16975 regfile6: 000000aa
3308 regfile7: 000000bb              16976 regfile7: 000000bb
3309 regfile8: 000000cc              16977 regfile8: 000000cc
3310 regfile9: 000000dd              16978 regfile9: 000000dd
3311 regfile10: 000000ee             16979 regfile10: 000000ee
3312 regfile11: 000000ff             16980 regfile11: 000000ff
3313 regfile12: 00000000             16981 regfile12: 00000000
3314 regfile13: 00000000             16982 regfile13: 00000000
3315 regfile14: 00000000             16983 regfile14: 00000000
3316 regfile15: 00000000             16984 regfile15: 00000000
3317 regfile16: 00000000             16985 regfile16: 00000000
3318 regfile17: 00000000             16986 regfile17: 00000000
3319 regfile18: 00000000             16987 regfile18: 00000000
3320 regfile19: 00000000             16988 regfile19: 00000000
3321 regfile20: 00000000             16989 regfile20: 00000000
3322 regfile21: 00000000             16990 regfile21: 00000000
3323 regfile22: 00000000             16991 regfile22: 00000000
3324 regfile23: 00000000             16992 regfile23: 00000000
3325 regfile24: 00000000             16993 regfile24: 00000000
3326 regfile25: 00000000             16994 regfile25: 00000000
3327 regfile26: 00000000             16995 regfile26: 00000000
3328 regfile27: 00000000             16996 regfile27: 00000000
3329 regfile28: 00000000             16997 regfile28: 00000000
3330 regfile29: 00000000             16998 regfile29: 00000000
3331 regfile30: 00000000             16999 regfile30: 00000000
3332 regfile31: 000f0000             17000 regfile31: 000f0000

+ File 821 lines unchanged, 13668 lines added, 2511 lines modified, 0 lines deleted.
```

LHU:

TextDiff

File Edit Search Options Actions Help

```
D:\cputest\38_lhu.result.txt          D:\cputest\my\my_38_lhu.txt
3301 regfile0: 00000000              16969 regfile0: 00000000
3302 regfile1: 0000001f              16970 regfile1: 00000001f
3303 regfile2: 00000003              16971 regfile2: 00000003
3304 regfile3: 00000004              16972 regfile3: 00000004
3305 regfile4: 00000005              16973 regfile4: 00000005
3306 regfile5: 00000006              16974 regfile5: 00000006
3307 regfile6: 00000007              16975 regfile6: 00000007
3308 regfile7: 00000008              16976 regfile7: 00000008
3309 regfile8: 00000009              16977 regfile8: 00000009
3310 regfile9: 0000000a              16978 regfile9: 0000000a
3311 regfile10: 0000000b             16979 regfile10: 0000000b
3312 regfile11: 0000000c             16980 regfile11: 0000000c
3313 regfile12: 0000000d             16981 regfile12: 0000000d
3314 regfile13: 0000000e             16982 regfile13: 0000000e
3315 regfile14: 0000000f             16983 regfile14: 0000000f
3316 regfile15: 00000010             16984 regfile15: 00000010
3317 regfile16: 00000011             16985 regfile16: 00000011
3318 regfile17: 00000012             16986 regfile17: 00000012
3319 regfile18: 00000013             16987 regfile18: 00000013
3320 regfile19: 00000014             16988 regfile19: 00000014
3321 regfile20: 00000015             16989 regfile20: 00000015
3322 regfile21: 00000016             16990 regfile21: 00000016
3323 regfile22: 00000017             16991 regfile22: 00000017
3324 regfile23: 00000018             16992 regfile23: 00000018
3325 regfile24: 00000019             16993 regfile24: 00000019
3326 regfile25: 0000001a             16994 regfile25: 0000001a
3327 regfile26: 0000001b             16995 regfile26: 0000001b
3328 regfile27: 0000001c             16996 regfile27: 0000001c
3329 regfile28: 0000001d             16997 regfile28: 0000001d
3330 regfile29: 0000001e             16998 regfile29: 0000001e
3331 regfile30: 00000000             16999 regfile30: 00000000
3332 regfile31: 0000001f             17000 regfile31: 0000001f

+ File 2653 lines unchanged, 13668 lines added, 679 lines modified, 0 lines deleted.
```

LHU2:

TextDiff

File Edit Search Options Actions Help

```
D:\cputest\38_lhu2.result.txt          D:\cputest\my\my_38_lhu2.txt
3301 regfile0: 00000000              16969 regfile0: 00000000
3302 regfile1: 00000000              16970 regfile1: 00000000
3303 regfile2: 0000001f              16971 regfile2: 000000ff
3304 regfile3: 0000ffff              16972 regfile3: 0000ffff
3305 regfile4: 0000ffff              16973 regfile4: 0000ffff
3306 regfile5: 00005555              16974 regfile5: 00005555
3307 regfile6: 0000aaaa              16975 regfile6: 0000aaaa
3308 regfile7: 0000bbbb              16976 regfile7: 0000bbbb
3309 regfile8: 0000cccc              16977 regfile8: 0000cccc
3310 regfile9: 0000dddd              16978 regfile9: 0000dddd
3311 regfile10: 0000eeee              16979 regfile10: 0000eeee
3312 regfile11: 0000ffff              16980 regfile11: 0000ffff
3313 regfile12: 00000000              16981 regfile12: 00000000
3314 regfile13: 00000000              16982 regfile13: 00000000
3315 regfile14: 00000000              16983 regfile14: 00000000
3316 regfile15: 00000000              16984 regfile15: 00000000
3317 regfile16: 00000000              16985 regfile16: 00000000
3318 regfile17: 00000000              16986 regfile17: 00000000
3319 regfile18: 00000000              16987 regfile18: 00000000
3320 regfile19: 00000000              16988 regfile19: 00000000
3321 regfile20: 00000000              16989 regfile20: 00000000
3322 regfile21: 00000000              16990 regfile21: 00000000
3323 regfile22: 00000000              16991 regfile22: 00000000
3324 regfile23: 00000000              16992 regfile23: 00000000
3325 regfile24: 00000000              16993 regfile24: 00000000
3326 regfile25: 00000000              16994 regfile25: 00000000
3327 regfile26: 00000000              16995 regfile26: 00000000
3328 regfile27: 00000000              16996 regfile27: 00000000
3329 regfile28: 00000000              16997 regfile28: 00000000
3330 regfile29: 00000000              16998 regfile29: 00000000
3331 regfile30: 00000000              16999 regfile30: 00000000
3332 regfile31: 000f0000              17000 regfile31: 000f0000

+ File 789 lines unchanged, 13668 lines added, 2543 lines modified, 0 lines deleted.
```

LHSH:

TextDiff

File Edit Search Options Actions Help

```
D:\cputest\40.41_lhsh.result.txt
```

```
3301 regfile0: 00000000
3302 regfile1: 0000001f
3303 regfile2: 00000003
3304 regfile3: 00000004
3305 regfile4: 00000005
3306 regfile5: 00000006
3307 regfile6: 00000007
3308 regfile7: 00000008
3309 regfile8: 00000009
3310 regfile9: 0000000a
3311 regfile10: 0000000b
3312 regfile11: 0000000c
3313 regfile12: 0000000d
3314 regfile13: 0000000e
3315 regfile14: 0000000f
3316 regfile15: 00000010
3317 regfile16: 00000011
3318 regfile17: 00000012
3319 regfile18: 00000013
3320 regfile19: 00000014
3321 regfile20: 00000015
3322 regfile21: 00000016
3323 regfile22: 00000017
3324 regfile23: 00000018
3325 regfile24: 00000019
3326 regfile25: 0000001a
3327 regfile26: 0000001b
3328 regfile27: 0000001c
3329 regfile28: 0000001d
3330 regfile29: 0000001e
3331 regfile30: 00000000
3332 regfile31: 0000001f
```

```
D:\cputest\my\my_40.41_lhsh.txt
```

```
16969 regfile0: 00000000
16970 regfile1: 00000000
16971 regfile2: 00000003
16972 regfile3: 00000004
16973 regfile4: 00000005
16974 regfile5: 00000006
16975 regfile6: 00000007
16976 regfile7: 00000008
16977 regfile8: 00000009
16978 regfile9: 0000000a
16979 regfile10: 0000000b
16980 regfile11: 0000000c
16981 regfile12: 0000000d
16982 regfile13: 0000000e
16983 regfile14: 0000000f
16984 regfile15: 00000010
16985 regfile16: 00000011
16986 regfile17: 00000012
16987 regfile18: 00000013
16988 regfile19: 00000014
16989 regfile20: 00000015
16990 regfile21: 00000016
16991 regfile22: 00000017
16992 regfile23: 00000018
16993 regfile24: 00000019
16994 regfile25: 0000001a
16995 regfile26: 0000001b
16996 regfile27: 0000001c
16997 regfile28: 0000001d
16998 regfile29: 0000001e
16999 regfile30: 00000000
17000 regfile31: 0000001f
```

+ Fil 789 lines unchanged, 13668 lines added, 2543 lines modified, 0 lines deleted.

LHSH2:

TextDiff

File Edit Search Options Actions Help

```
D:\cputest\40.41_lhsh2.result.txt
```

```
3301 regfile0: 00000000
3302 regfile1: 00000000
3303 regfile2: 0000001f
3304 regfile3: 0000ffff
3305 regfile4: ffffffff
3306 regfile5: 00005555
3307 regfile6: fffffaaa
3308 regfile7: fffffbbb
3309 regfile8: fffffccc
3310 regfile9: fffffddd
3311 regfile10: fffffeee
3312 regfile11: ffffffff
3313 regfile12: 00000000
3314 regfile13: 00000000
3315 regfile14: 00000000
3316 regfile15: 00000000
3317 regfile16: 00000000
3318 regfile17: 00000000
3319 regfile18: 00000000
3320 regfile19: 00000000
3321 regfile20: 00000000
3322 regfile21: 00000000
3323 regfile22: 00000000
3324 regfile23: 00000000
3325 regfile24: 00000000
3326 regfile25: 00000000
3327 regfile26: 00000000
3328 regfile27: 00000000
3329 regfile28: 00000000
3330 regfile29: 00000000
3331 regfile30: 00000000
3332 regfile31: 000f0000
```

```
D:\cputest\my\my_40.41_lhsh2.txt
```

```
16969 regfile0: 00000000
16970 regfile1: 00000000
16971 regfile2: 000000ff
16972 regfile3: 0000ffff
16973 regfile4: ffffffff
16974 regfile5: 00005555
16975 regfile6: fffffaaa
16976 regfile7: fffffbbb
16977 regfile8: fffffccc
16978 regfile9: fffffddd
16979 regfile10: fffffeee
16980 regfile11: ffffffff
16981 regfile12: 00000000
16982 regfile13: 00000000
16983 regfile14: 00000000
16984 regfile15: 00000000
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 00000000
16988 regfile19: 00000000
16989 regfile20: 00000000
16990 regfile21: 00000000
16991 regfile22: 00000000
16992 regfile23: 00000000
16993 regfile24: 00000000
16994 regfile25: 00000000
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 00000000
16999 regfile30: 00000000
17000 regfile31: 000f0000
```

+ Fil 2623 lines unchanged, 13668 lines added, 709 lines modified, 0 lines deleted.

MFCOMTC0:

TextDiff

File Edit Search Options Actions Help

D:\cputest\42.45_mfc0mtc0.result.txt D:\cputest\my\my_42.45_mfc0mtc0.txt

```

411 regfile0: 00000000
412 regfile1: 00000003
413 regfile2: ffffffff
414 regfile3: 00005555
415 regfile4: fffffaaa
416 regfile5: 00000000
417 regfile6: 00000000
418 regfile7: 00000000
419 regfile8: 00000003
420 regfile9: 00000000
421 regfile10: 00000000
422 regfile11: 00000000
423 regfile12: ffffffff
424 regfile13: 00005555
425 regfile14: fffffaaa
426 regfile15: 00000000
427 regfile16: 00000000
428 regfile17: 00000000
429 regfile18: 00000000
430 regfile19: 00000000
431 regfile20: 00000000
432 regfile21: 00000000
433 regfile22: 00000000
434 regfile23: 00000000
435 regfile24: 00000000
436 regfile25: 00000000
437 regfile26: 00000000
438 regfile27: 00000000
439 regfile28: 00000000
440 regfile29: 00000000
441 regfile30: 00000000
442 regfile31: 00000000

16969 regfile0: 00000000
16970 regfile1: 00000003
16971 regfile2: ffffffff
16972 regfile3: 00005555
16973 regfile4: fffffaaa
16974 regfile5: 00000000
16975 regfile6: 00000000
16976 regfile7: 00000000
16977 regfile8: 00000003
16978 regfile9: 00000000
16979 regfile10: 00000000
16980 regfile11: 00000000
16981 regfile12: ffffffff
16982 regfile13: 00005555
16983 regfile14: fffffaaa
16984 regfile15: 00000000
16985 regfile16: 00000000
16986 regfile17: 00000000
16987 regfile18: 00000000
16988 regfile19: 00000000
16989 regfile20: 00000000
16990 regfile21: 00000000
16991 regfile22: 00000000
16992 regfile23: 00000000
16993 regfile24: 00000000
16994 regfile25: 00000000
16995 regfile26: 00000000
16996 regfile27: 00000000
16997 regfile28: 00000000
16998 regfile29: 00000000
16999 regfile30: 00000000
17000 regfile31: 00000000

```

+ - 374 lines unchanged, 16558 lines added, 68 lines modified, 0 lines deleted.

MFHIMTHI:

TextDiff

File Edit Search Options Actions Help

D:\cputest\43.46_mfhimthi.result.txt D:\cputest\my\my_43.46_mfhimthi.txt

```

615 regfile0: 00000000
616 regfile1: 00005555
617 regfile2: 00000001
618 regfile3: ffffffff
619 regfile4: 0000000f
620 regfile5: 00000000
621 regfile6: 0000aaaa
622 regfile7: 00005555
623 regfile8: 00000000
624 regfile9: 00000000
625 regfile10: 00000000
626 regfile11: 00000000
627 regfile12: 00000000
628 regfile13: 00000000
629 regfile14: 00000000
630 regfile15: 00000000
631 regfile16: 00000000
632 regfile17: 00000000
633 regfile18: 00000000
634 regfile19: 00000000
635 regfile20: 00000000
636 regfile21: 00000000
637 regfile22: 00000000
638 regfile23: 00000000
639 regfile24: 00000000
640 regfile25: 00000000
641 regfile26: 00000000
642 regfile27: 00000000
643 regfile28: 00000000
644 regfile29: 00000000
645 regfile30: 00000000
646 regfile31: 00000000

40769 regfile0: 00000000
40770 regfile1: 00005555
40771 regfile2: 00000001
40772 regfile3: ffffffff
40773 regfile4: 0000000f
40774 regfile5: 00000000
40775 regfile6: 0000aaaa
40776 regfile7: 00005555
40777 regfile8: 00000000
40778 regfile9: 00000000
40779 regfile10: 00000000
40780 regfile11: 00000000
40781 regfile12: 00000000
40782 regfile13: 00000000
40783 regfile14: 00000000
40784 regfile15: 00000000
40785 regfile16: 00000000
40786 regfile17: 00000000
40787 regfile18: 00000000
40788 regfile19: 00000000
40789 regfile20: 00000000
40790 regfile21: 00000000
40791 regfile22: 00000000
40792 regfile23: 00000000
40793 regfile24: 00000000
40794 regfile25: 00000000
40795 regfile26: 00000000
40796 regfile27: 00000000
40797 regfile28: 00000000
40798 regfile29: 00000000
40799 regfile30: 00000000
40800 regfile31: 00000000

```

+ - 597 lines unchanged, 40154 lines added, 49 lines modified, 0 lines deleted.

MFLOMCLO:

TextDiff

File Edit Search Options Actions Help

```
D:\cputest\my\my_44.47_mflo.mtio.txt
40769 regfile0: 00000000
40770 regfile1: 00005555
40771 regfile2: 00000001
40772 regfile3: ffffffff
40773 regfile4: 0000000f
40774 regfile5: 00000000
40775 regfile6: 0000aaaa
40776 regfile7: 00005555
40777 regfile8: 00000000
40778 regfile9: 00000000
40779 regfile10: 00000000
40780 regfile11: 00000000
40781 regfile12: 00000000
40782 regfile13: 00000000
40783 regfile14: 00000000
40784 regfile15: 00000000
40785 regfile16: 00000000
40786 regfile17: 00000000
40787 regfile18: 00000000
40788 regfile19: 00000000
40789 regfile20: 00000000
40790 regfile21: 00000000
40791 regfile22: 00000000
40792 regfile23: 00000000
40793 regfile24: 00000000
40794 regfile25: 00000000
40795 regfile26: 00000000
40796 regfile27: 00000000
40797 regfile28: 00000000
40798 regfile29: 00000000
40799 regfile30: 00000000
40800 regfile31: 00000000

D:\cputest\my\my_44.47_mflo.mtio.result.txt
615 regfile0: 00000000
616 regfile1: 00005555
617 regfile2: 00000001
618 regfile3: ffffffff
619 regfile4: 0000000f
620 regfile5: 00000000
621 regfile6: 0000aaaa
622 regfile7: 00005555
623 regfile8: 00000000
624 regfile9: 00000000
625 regfile10: 00000000
626 regfile11: 00000000
627 regfile12: 00000000
628 regfile13: 00000000
629 regfile14: 00000000
630 regfile15: 00000000
631 regfile16: 00000000
632 regfile17: 00000000
633 regfile18: 00000000
634 regfile19: 00000000
635 regfile20: 00000000
636 regfile21: 00000000
637 regfile22: 00000000
638 regfile23: 00000000
639 regfile24: 00000000
640 regfile25: 00000000
641 regfile26: 00000000
642 regfile27: 00000000
643 regfile28: 00000000
644 regfile29: 00000000
645 regfile30: 00000000
646 regfile31: 00000000

< > < > + - 597 lines unchanged, 40154 lines added, 49 lines modified, 0 lines deleted.
```

MUL:

TextDiff

File Edit Search Options Actions Help

```
D:\cputest\my\my_48_mul.txt
16969 regfile0: 00000000
16970 regfile1: fffffef0
16971 regfile2: 0000004e
16972 regfile3: fffffef0
16973 regfile4: 0000004e
16974 regfile5: fffffef0
16975 regfile6: 0000004e
16976 regfile7: fffffef0
16977 regfile8: 0000004e
16978 regfile9: fffffef0
16979 regfile10: 0000004e
16980 regfile11: fffffef0
16981 regfile12: 0000004e
16982 regfile13: fffffef0
16983 regfile14: 0000004e
16984 regfile15: fffffef0
16985 regfile16: 0000004e
16986 regfile17: fffffef0
16987 regfile18: 0000004e
16988 regfile19: fffffef0
16989 regfile20: 0000004e
16990 regfile21: fffffef0
16991 regfile22: 0000004e
16992 regfile23: fffffef0
16993 regfile24: 0000004e
16994 regfile25: fffffef0
16995 regfile26: 0000004e
16996 regfile27: fffffef0
16997 regfile28: 0000004e
16998 regfile29: fffffef0
16999 regfile30: fffffef0
17000 regfile31: 0000004e

D:\cputest\my\my_48_mul.result.txt
9251 regfile0: 00000000
9252 regfile1: fffffef0
9253 regfile2: 0000004e
9254 regfile3: fffffef0
9255 regfile4: 0000004e
9256 regfile5: fffffef0
9257 regfile6: 0000004e
9258 regfile7: fffffef0
9259 regfile8: 0000004e
9260 regfile9: fffffef0
9261 regfile10: 0000004e
9262 regfile11: fffffef0
9263 regfile12: 0000004e
9264 regfile13: fffffef0
9265 regfile14: 0000004e
9266 regfile15: fffffef0
9267 regfile16: 0000004e
9268 regfile17: fffffef0
9269 regfile18: 0000004e
9270 regfile19: fffffef0
9271 regfile20: 0000004e
9272 regfile21: fffffef0
9273 regfile22: 0000004e
9274 regfile23: fffffef0
9275 regfile24: 0000004e
9276 regfile25: fffffef0
9277 regfile26: 0000004e
9278 regfile27: fffffef0
9279 regfile28: 0000004e
9280 regfile29: fffffef0
9281 regfile30: fffffef0
9282 regfile31: 0000004e

< > < > + - 8838 lines unchanged, 7718 lines added, 444 lines modified, 0 lines deleted.
```

MULTU:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_49_multu.txt

```

9251 regfile0: 00000000
9252 regfile1: ffffffc8
9253 regfile2: 0000004e
9254 regfile3: ffffffc8
9255 regfile4: 0000004e
9256 regfile5: ffffffc8
9257 regfile6: 0000004e
9258 regfile7: ffffffc8
9259 regfile8: 0000004e
9260 regfile9: ffffffc8
9261 regfile10: 0000004e
9262 regfile11: ffffffc8
9263 regfile12: 0000004e
9264 regfile13: ffffffc8
9265 regfile14: 0000004e
9266 regfile15: ffffffc8
9267 regfile16: 0000004e
9268 regfile17: ffffffc8
9269 regfile18: 0000004e
9270 regfile19: ffffffc8
9271 regfile20: 0000004e
9272 regfile21: ffffffc8
9273 regfile22: 0000004e
9274 regfile23: ffffffc8
9275 regfile24: 0000004e
9276 regfile25: ffffffc8
9277 regfile26: 0000004e
9278 regfile27: ffffffc8
9279 regfile28: 0000004e
9280 regfile29: ffffffc8
9281 regfile30: ffffffc8
9282 regfile31: 0000004e

```

D:\cputest\my\my_49_multu.result.txt

```

40769 regfile0: 00000000
40770 regfile1: ffffffc8
40771 regfile2: 0000004e
40772 regfile3: ffffffc8
40773 regfile4: 0000004e
40774 regfile5: ffffffc8
40775 regfile6: 0000004e
40776 regfile7: ffffffc8
40777 regfile8: 0000004e
40778 regfile9: ffffffc8
40779 regfile10: 0000004e
40780 regfile11: ffffffc8
40781 regfile12: 0000004e
40782 regfile13: ffffffc8
40783 regfile14: 0000004e
40784 regfile15: ffffffc8
40785 regfile16: 0000004e
40786 regfile17: ffffffc8
40787 regfile18: 0000004e
40788 regfile19: ffffffc8
40789 regfile20: 0000004e
40790 regfile21: ffffffc8
40791 regfile22: 0000004e
40792 regfile23: ffffffc8
40793 regfile24: 0000004e
40794 regfile25: ffffffc8
40795 regfile26: 0000004e
40796 regfile27: ffffffc8
40797 regfile28: 0000004e
40798 regfile29: ffffffc8
40799 regfile30: ffffffc8
40800 regfile31: 0000004e

```

+ - 9091 lines unchanged, 31518 lines added, 191 lines modified, 0 lines deleted.

BGEZ:

TextDiff

File Edit Search Options Actions Help

D:\cputest\my\my_52_bgez.txt

```

411 regfile0: 00000000
412 regfile1: 00000000
413 regfile2: 00000002
414 regfile3: 00000002
415 regfile4: 00000002
416 regfile5: 00000002
417 regfile6: 00000000
418 regfile7: 00000000
419 regfile8: 00000000
420 regfile9: 00000000
421 regfile10: ffffffff
422 regfile11: 00000000
423 regfile12: 00000000
424 regfile13: 00000000
425 regfile14: 00000000
426 regfile15: 00000000
427 regfile16: 00000000
428 regfile17: 00000000
429 regfile18: 00000000
430 regfile19: 00000000
431 regfile20: 00000001
432 regfile21: 00000000
433 regfile22: 00000000
434 regfile23: 00000000
435 regfile24: 00000000
436 regfile25: 00000000
437 regfile26: 00000000
438 regfile27: 00000000
439 regfile28: 00000000
440 regfile29: 00000000
441 regfile30: 00000000
442 regfile31: 00000000

```

D:\cputest\my\my_52_bgez.result.txt

```

40769 regfile0: 00000000
40770 regfile1: 00000000
40771 regfile2: 00000002
40772 regfile3: 00000002
40773 regfile4: 00000002
40774 regfile5: 00000002
40775 regfile6: 00000000
40776 regfile7: 00000000
40777 regfile8: 00000000
40778 regfile9: 00000000
40779 regfile10: ffffffff
40780 regfile11: 00000000
40781 regfile12: 00000000
40782 regfile13: 00000000
40783 regfile14: 00000000
40784 regfile15: 00000000
40785 regfile16: 00000000
40786 regfile17: 00000000
40787 regfile18: 00000000
40788 regfile19: 00000000
40789 regfile20: 00000001
40790 regfile21: 00000000
40791 regfile22: 00000000
40792 regfile23: 00000000
40793 regfile24: 00000000
40794 regfile25: 00000000
40795 regfile26: 00000000
40796 regfile27: 00000000
40797 regfile28: 00000000
40798 regfile29: 00000000
40799 regfile30: 00000000
40800 regfile31: 00000000

```

+ - 434 lines unchanged, 40358 lines added, 8 lines modified, 0 lines deleted.

DIV:

```

D:\cputest\my\my_54_div.txt
40769 regfile0: 00000000
40770 regfile1: 00000000
40771 regfile2: 00000002
40772 regfile3: ffffffa
40773 regfile4: ffffffd
40774 regfile5: ffffffa
40775 regfile6: ffffffd
40776 regfile7: ffffffa
40777 regfile8: ffffffd
40778 regfile9: ffffffa
40779 regfile10: ffffffd
40780 regfile11: ffffffa
40781 regfile12: ffffffd
40782 regfile13: ffffffa
40783 regfile14: ffffffd
40784 regfile15: ffffffa
40785 regfile16: ffffffd
40786 regfile17: ffffffa
40787 regfile18: ffffffd
40788 regfile19: ffffffa
40789 regfile20: ffffffd
40790 regfile21: ffffffa
40791 regfile22: ffffffd
40792 regfile23: ffffffa
40793 regfile24: ffffffd
40794 regfile25: ffffffa
40795 regfile26: ffffffd
40796 regfile27: ffffffa
40797 regfile28: ffffffd
40798 regfile29: ffffffa
40799 regfile30: ffffffd
40800 regfile31: ffffffd

```

+ ~ - 14182 lines unchanged, 26078 lines added, 540 lines modified, 0 lines deleted.

2、静态流水线的仿真过程

1.1 Testbench 程序

```
'timescale 1ns / 1ps
```

```

module cpu_tb();
    reg clk;
    reg rst;
    wire [7:0] o_seg;
    wire [7:0] o_sel;

    //看控制信号
    wire [31:0]id_pc_in = uut.instdecode.id_pc_in;
    wire [31:0]id_inst_in = uut.instdecode.id_inst_in;
    wire [31:0]stage_id_inst = uut.stage_id_inst;
    wire [31:0]stage_exe_inst = uut.stage_exe_inst;
    //    wire [31:0]stage_mem_inst = uut.stage_mem_inst;

    wire id_stall_out = uut.instdecode.id_stall_out;
    wire [1:0]stall_count = uut.instdecode.stall_count;

```

```

wire isbranch = uut.isbranch;
wire [31:0]branch_pc = uut.branch_pc;

wire [31:0]id_alu_a = uut.instdecode.id_alu_a;
wire [31:0]id_alu_b = uut.instdecode.id_alu_b;
wire [2:0]alu_achoice = uut.instdecode.alu_achoice;
wire [2:0]alu_bchoice = uut.instdecode.alu_bchoice;
wire [4:0]id_aluchoice = uut.instdecode.id_aluchoice;
wire [31:0]hi_result = uut.exe_alu_hiout;
wire [31:0]ans = uut.ans;

wire [31:0]reg28 = uut.instdecode.cpu_ref.array_reg[28];
wire [31:0]reg2 = uut.instdecode.cpu_ref.array_reg[2];
wire [31:0]reg3 = uut.instdecode.cpu_ref.array_reg[3];

sccomp_dataflow uut(
    .clk_in(clk),
    .reset(rst),
    .o_seg(o_seg),
    .o_sel(o_sel)
);
integer file_output;
integer counter;
initial
begin
    file_output = $fopen("D:/cputest/my/my_1.txt");//可根据需要调整
    clk = 1;
    rst = 1;
    counter = 0;
    #4;
    rst = 0;
end

always
begin
#1;
clk = ~clk;
if (clk == 1'b1)
begin
if (counter == 200000)
begin
    $fclose(file_output);
end
else begin

```

```

counter = counter + 1;
if(counter<200000)
begin
$fdisplay(file_output,"pc: %h",uut.stage_if_pc);
$fdisplay(file_output,"instr: %h",uut.stage_if_inst);
$fdisplay(file_output,"regfile0: %h",uut.instdecode.cpu_ref.array_reg[0]);
$fdisplay(file_output,"regfile1: %h",uut.instdecode.cpu_ref.array_reg[1]);
$fdisplay(file_output,"regfile2: %h",uut.instdecode.cpu_ref.array_reg[2]);
$fdisplay(file_output,"regfile3: %h",uut.instdecode.cpu_ref.array_reg[3]);
$fdisplay(file_output,"regfile4: %h",uut.instdecode.cpu_ref.array_reg[4]);
$fdisplay(file_output,"regfile5: %h",uut.instdecode.cpu_ref.array_reg[5]);
$fdisplay(file_output,"regfile6: %h",uut.instdecode.cpu_ref.array_reg[6]);
$fdisplay(file_output,"regfile7: %h",uut.instdecode.cpu_ref.array_reg[7]);
$fdisplay(file_output,"regfile8: %h",uut.instdecode.cpu_ref.array_reg[8]);
$fdisplay(file_output,"regfile9: %h",uut.instdecode.cpu_ref.array_reg[9]);
$fdisplay(file_output,"regfile10: %h",uut.instdecode.cpu_ref.array_reg[10]);
$fdisplay(file_output,"regfile11: %h",uut.instdecode.cpu_ref.array_reg[11]);
$fdisplay(file_output,"regfile12: %h",uut.instdecode.cpu_ref.array_reg[12]);
$fdisplay(file_output,"regfile13: %h",uut.instdecode.cpu_ref.array_reg[13]);
$fdisplay(file_output,"regfile14: %h",uut.instdecode.cpu_ref.array_reg[14]);
$fdisplay(file_output,"regfile15: %h",uut.instdecode.cpu_ref.array_reg[15]);
$fdisplay(file_output,"regfile16: %h",uut.instdecode.cpu_ref.array_reg[16]);
$fdisplay(file_output,"regfile17: %h",uut.instdecode.cpu_ref.array_reg[17]);
$fdisplay(file_output,"regfile18: %h",uut.instdecode.cpu_ref.array_reg[18]);
$fdisplay(file_output,"regfile19: %h",uut.instdecode.cpu_ref.array_reg[19]);
$fdisplay(file_output,"regfile20: %h",uut.instdecode.cpu_ref.array_reg[20]);
$fdisplay(file_output,"regfile21: %h",uut.instdecode.cpu_ref.array_reg[21]);
$fdisplay(file_output,"regfile22: %h",uut.instdecode.cpu_ref.array_reg[22]);
$fdisplay(file_output,"regfile23: %h",uut.instdecode.cpu_ref.array_reg[23]);
$fdisplay(file_output,"regfile24: %h",uut.instdecode.cpu_ref.array_reg[24]);
$fdisplay(file_output,"regfile25: %h",uut.instdecode.cpu_ref.array_reg[25]);
$fdisplay(file_output,"regfile26: %h",uut.instdecode.cpu_ref.array_reg[26]);
$fdisplay(file_output,"regfile27: %h",uut.instdecode.cpu_ref.array_reg[27]);
$fdisplay(file_output,"regfile28: %h",uut.instdecode.cpu_ref.array_reg[28]);
$fdisplay(file_output,"regfile29: %h",uut.instdecode.cpu_ref.array_reg[29]);
$fdisplay(file_output,"regfile30: %h",uut.instdecode.cpu_ref.array_reg[30]);
$fdisplay(file_output,"regfile31: %h",uut.instdecode.cpu_ref.array_reg[31]);
end
end
end
endmodule

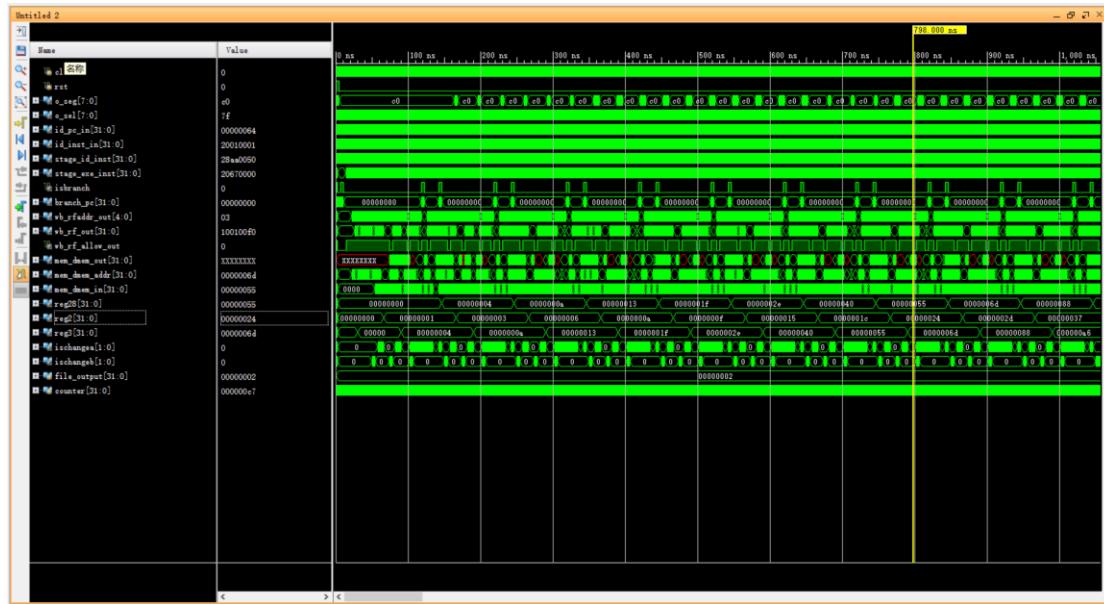
```

1.2 测试情况

在静态流水线实验中，同动态流水线一样，我采用之前计算机组原理设计课上提供的 54 条指令 CPU 的测试文件，将 pc、指令以及寄存器的值输出到 txt 文件中进行了测试；由于流水线 CPU 无法在下一条指令取指前写回寄存器，通过观察最终的寄存器情况来判定 CPU 设计是否正确。测试情况在之前的静态流水线实验中已经展示过，在此不再赘述。

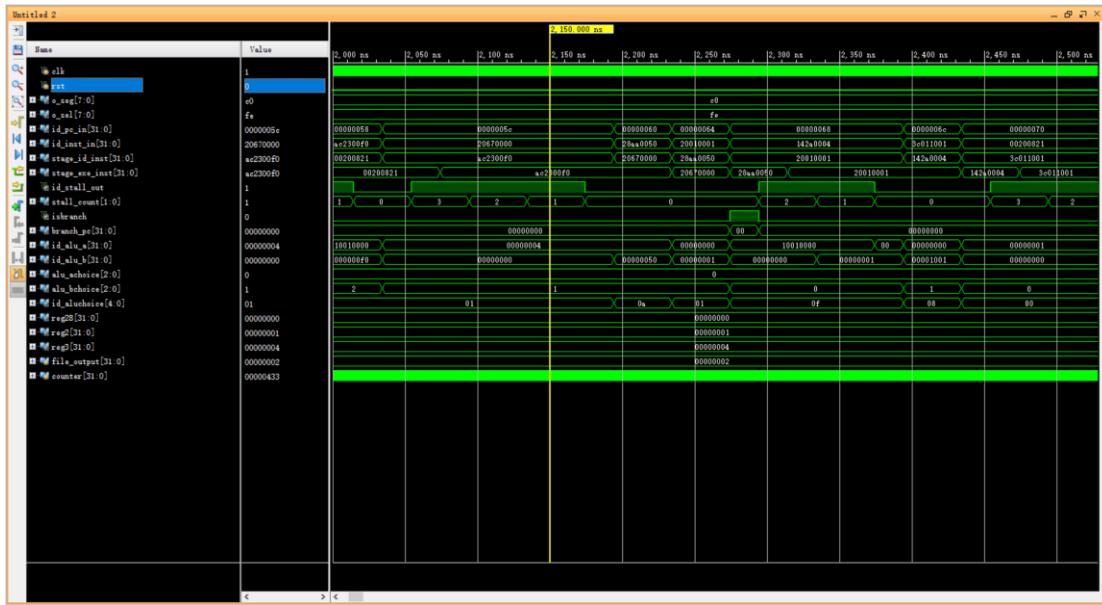
五、实验仿真的波形图及某时刻寄存器值的物理意义

1、动态流水线的波形图及某时刻寄存器值的物理意义



在上面的波形图中，重点观察 reg2、reg3、reg28 三个寄存器，其中 reg2 代表实验算数学模型中的 a，reg3 代表实验算数学模型中的 b，reg28 代表实验算数学模型中的 d，在图示中，可以看到 reg2 的变化为十六进制下的 0、1、3、6、a、f、15、1c、24、2d、37 等，符合数学模型中 a 的变化。reg3 的变化为十六进制下的 0、4、a、13、1f、2e、40、55、6d、88、a6 等，符合数学模型中 b 的变化。reg28 的变化为十六进制下的 0、4、a、13、1f、2e、40、55、6d、88 等，符合数学模型中 d 的变化。

2、静态流水线的波形图及某时刻寄存器值的物理意义



在上面的波形图中，重点观察 reg2、reg3、reg28 三个寄存器，其中 reg2 代表实验算数学模型中的 a，reg3 代表实验算数学模型中的，reg28 代表实验算数学模型中的 d，在图示当前时刻，可以看到 a=1，b=4，d=0，这是因为当前 a[1]、b[1]都已经计算出来，但还没有执行到计算 d 以及写入 reg28 的指令。

六、实验算数学模型及算法程序

1、数学模型

```

int a[m],b[m],c[m],d[m];
a[0]=0; +
b[0]=1; +
a[i]=a[i-1]+i; +
b[i]=b[i-1]+3i; +
c[i]=  

  { a[i],      0≤i≤19  

  { a[i]+b[i], 20≤i≤39  

  { a[i]*b[i], 40≤i≤59  

+
d[i]=  

  { b[i],      0≤i≤19  

  { a[i]*c[i], 20≤i≤39  

  { c[i]*b[i], 40≤i≤59
  
```

2、算法汇编程序

```
.data
A:.space 240
B:.space 240
C:.space 240
D:.spcce 240

.text
j main
exc:
nop
j exc

main:
addi $2,$0,0 #a[i]
addi $3,$0,1 #b[i]
addi $4,$0,0
addi $5,$0,4 #counter
addi $6,$0,0 #a[i-1]
addi $7,$0,1 #b[i-1]
addi $8,$0,0xC #counter*3
addi $10,$0,0
addi $11,$0,240

loop:
srl $12,$5,2
add $2,$6,$12 #a[i]=a[i-1]+i
sw $2, A($0)
addi $6,$2,0 #a[i-1]=a[i]

srl $12,$8,2
add $3,$7,$12 #b[i]=b[i-1]+i*3
sw $3, B($0)
addi $7,$3,0 #b[i-1]=b[i]

slt $10,$5,80 #if counter<80 $10=1
addi $1,$0,0x0001 #$1 涓哄颤绉诲湴鎬?
bne $1,$10,c1
sw $3,D($0)
j endd

c1:
```

```
slti $10,$5,160 #if counter<160 $10=1  
addi $1,$0,0x0001  
bne $1,$10,c2  
addu $4,$2,$3  
mul $4,$2,$4  
sw $4,D($0)  
j endd
```

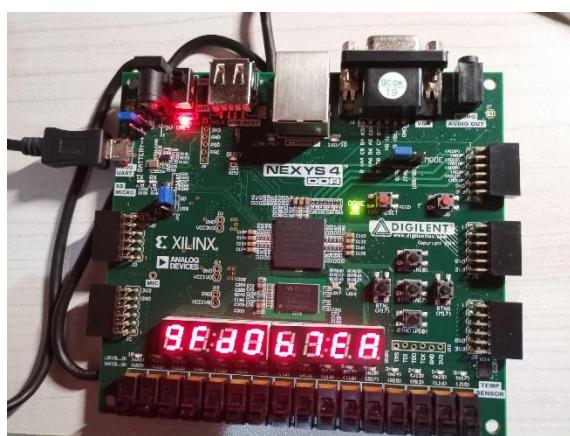
```
c2:  
addi $4,$2,0  
mul $4,$3,$4  
mul $4,$3,$4  
sw $4,D($0)
```

```
endd:  
lw $28,D($0)  
addi $5,$5,4  
addi $8,$8,0xC  
bne $5,$11,loop
```

```
j exc
```

七、实验验算程序下板测试过程与实现

经仿真、综合、实现、生成比特流、比特流文件写入等步骤，最终下板结果如下；



可看到 d[59]的结果正确。

八、流水线的性能指标定性分析（包括：吞吐率、加速比、效率及相关与冲突分析）

1、动态流水线的性能指标定性分析

1.1 数据相关

对于静态流水线 CPU，可能会出现两种数据相关，第一种是前一条指令的目的操作数是下一条指令的源操作数，第二种是多条指令数据相关。

对于第一种情况，我们采用内部前推的解决方法，将 EXE 级和 MEM 级的输出结果前推到 ID 级，在 ID 级进行数据传递的判断。

对于第二种情况，我们采取暂停流水线的方法消除数据相关引起的冲突。

1.2 控制相关

对于 J、JL、JR、JALR、BEQ、BNE、BGEZ、TEQ 这几条分支或跳转指令，采用延迟槽的思想，并将分支判断前移到 ID 阶段，使得分支延迟为一个周期。

当分支成功或转移发生时，令处于延迟槽内的指令失效，即封锁其写信号，若延迟槽内也为分支指令，则禁止其对 PC 的修改。

当分支失败时，延迟槽内的指令正常执行。

1.3 冲突

1) 数据冲突：

可利用内部前推或者暂停流水线消除冲突。

2) 控制冲突：

可利用延迟槽和提前分支判断消除冲突。

3) 同时读写寄存器堆冲突：

在出现这样的冲突时，前半周期读、后半周期写的方法并不能解决问题。于是，在寄存器堆中添加三目运算符，若需要同时读写同一单元时，则将写入的值直接作为读出的值，便可解决该冲突。

1.4 吞吐率

完成计算任务总共执行了 1685 条指令，无失效延迟槽指令，无 stall 指令，通过和排空需要 4 个时钟周期，所以：

吞吐率=(1685)/(1685+4)=0.998/时钟周期。

1.5 加速比

完成这 1685 条指令理论所用时间为 $198*2+1369*4+118*5=6462$ 个时钟周期，实际所用时间为 1689 个时钟周期，所以

$$\text{加速比} = 6462/1689 = 3.826。$$

1.6 效率

完成计算任务总共执行了 1710 条指令，所以

$$\text{效率} = 198*2+1369*4+118*5/1710*5 = 75.6\%。$$

2、静态流水线的性能指标定性分析

2.1 数据相关

对于静态流水线 CPU，可能会出现两种数据相关，第一种是前一条指令的目的操作数是下一条指令的源操作数，第二种是多条指令数据相关，对于这两种情况，我们都采取暂停流水线的方法消除数据相关引起的冲突。

2.2 控制相关

对于 J、JL、JR、JALR、BEQ、BNE、BGEZ、TEQ 这几条分支或跳转指令，采用延迟槽的思想，并将分支判断前移到 ID 阶段，使得分支延迟为一个周期。

当分支成功或转移发生时，令处于延迟槽内的指令失效，即封锁其写信号，若延迟槽内也为分支指令，则禁止其对 PC 的修改。

当分支失败时，延迟槽内的指令正常执行。

2.3 冲突

1) 数据冲突：

可利用暂停流水线消除冲突。

2) 控制冲突：

可利用延迟槽和提前分支判断消除冲突。

3) 同时读写寄存器堆冲突：

在出现这样的冲突时，前半周期读、后半周期写的方法并不能解决问题。于是，在寄存器堆中添加三目运算符，若需要同时读写同一单元时，则将写入的值直接作为读出的值，便可解决该冲突。

2.4 吞吐率

完成计算任务总共执行了 1805 条指令，无失效的延迟槽指令，120 条 stall 指令，通过和排空需要 4 个时钟周期，所以：

$$\text{吞吐率} = (1805) / (1805 + 4 + 120 * 2) = 0.880/\text{时钟周期}。$$

2.5 加速比

完成这 1805 条指令理论所用时间为 $198 * 2 + 1369 * 4 + 118 * 5 + 120 * 1 = 6582$ 个时钟周期，实际所用时间为 2049 个时钟周期，所以

$$\text{加速比} = 6582 / 2049 = 3.212。$$

2.6 效率

完成计算任务总共执行了 1805 条指令，120 条 stall 指令，所以

$$\text{效率} = 198 * 2 + 1369 * 4 + 118 * 5 + 120 * 1 / 1805 * 5 = 72.9\%。$$

3、两者性能差异分析

3.1 性能差异原因

动态流水线和静态流水线相比，产生性能差异的主要原因就是动态流水线采用了内部前推技术，使得 stall 的情况大幅减少，因此提升了性能，下面从吞吐率、加速比、效率三方面来比较动态流水线和静态流水线的性能。

3.2 吞吐率

动态流水线吞吐率为 0.998/时钟周期，静态流水线吞吐率为 0.880/时钟周期，因此动态流水线的吞吐率更大。

3.3 加速比

动态流水线加速比为 3.826，静态流水线加速比为 3.212，因此动态流水线的加速比更好。

3.4 效率

动态流水线效率为 75.6%，静态流水线效率为 72.9%，因此动态流水线的效率更高。

九、总结与体会

1、总结

经查看波形图，输出 PC、指令及寄存器到 txt 文件以及下板测试，使用 Verilog 语言设计出的 54 条指令动态流水线 CPU 能够对指令正确执行与输出，本次实验成功。

2、体会

对于本次实验，首先的一点感受就是对于 Verilog 语言的理解更加深刻了。由于动态流水线 CPU 中涉及到很多控制信号、寄存器等的操作，因此，直接采用高级语言程序设计的思想去用 Verilog 语言进行底层逻辑电路的设计是完全行不通的。

理解 Verilog 语言最关键的一点在于代码与底层逻辑电路的对应关系，只要理解了代码与底层逻辑电路的对应关系，在进行逻辑设计的时候才能够尽可能地去避免一些细节上的错误，其中最简单最基本的一个例子就是，某个东西在不被驱动后，值是否会发生变化，这也就是我们经常提到的 wire 与 reg 的一个区别，这个区别对应到底层电路上就是导线与寄存器的区别，这就可以理解这个问题了。

当然，这只是一个最最简单的例子，理解 Verilog 对应的底层逻辑电路，还有一个很重要的应用就是 CPU 性能的分析。我们都知道，CPU 的性能主要是受门级电路的影响，那么，门的级数有时候就可以看作是一个比较合理的衡量标准，若存在两个 I/O 关系等价的逻辑模块，那么，究竟选择哪个模块就要考虑其在底层电路中对应的门的级数，选择对应门级数少的模块，这种模块会使 CPU 效率更高，在实验中如果在某些模块例如 controlunit，采用 assign 描述而不是采用 always 语句描述就可以减少最终生成的电路的门的数量，因此在本次实验中，我对之前静态流水线的许多模块进行了重构。

另外，理解动态流水线 CPU 的基本原理与结构也是本实验的重要目的之一。之前对于流水线结构的掌握与学习都是在理论层面的学习，虽然对于流水线的底层结构与器件布局有了一些了解，并懂得了指令执行的基本流程以及 CPU 的大致运行原理，但是，能力也仅仅局限于能看懂并理解课本上的图是什么意思，数据是怎么流通的，程序的怎么执行的等等，而对于从 0 开始设计一个流水线 CPU 却还是感觉摸不着头脑，对流水线 CPU 的结构的理解还是无法达到透彻理解的水平。

通过这次实验设计，我对流水线 CPU 后有了更加深入的理解，从原本的静态流水线到升级为一个动态流水线 CPU 架构，再到部分模块的重构，有了一个更加深入的 CPU 设计思路。并且，有时候也考虑了一些优化方法的优缺点，比如模块是否能简单集成使 CPU 的结构更加清晰，是否会影响 CPU 效率，以及对各种控制信号去进行协调能否稳定等，这都是在本实验中需要考虑到的问题，混乱的 CPU 架构不仅会导致容易出现错误，并且代码质量过低也会导致在开发的过程中进度缓慢，加之也不利于单个模块的测试。这次试验使得我对流水线 CPU 的基本原理有了更深刻的理解，对于每个部件的功能、指令的流程等都有了更深入的探究。

本次实验也大大增加了对 CPU 甚至计算机系统结构的兴趣。其实本来就对于计算机硬件比较感兴趣，并一直有志于为国产 CPU 的发展贡献出自己的一份力。在这次实验中，通过一条一条指令的分析与实现，使我感觉到 CPU 设计过程的乐趣，并对于 CPU 性能面临的问题与解决方案有了一个基本的了解。这也正是 CPU 设计过程中一个非常有趣的地方。虽

然花费了不少的时间，但是有着巨大的收获，非常值得。

十、附件（所有程序）

1、动态流水线的设计程序

1.1 sccomp_dataflow

```
'timescale 1ns / 1ps
module sccomp_dataflow(
    input clk, //时钟
    input reset, //复位
    output [7:0] o_seg,
    output [7:0] o_sel
);
wire clk_in;
clk_wiz myclk(
    clk,
    reset,
    clk_in
);

wire [31:0]ans;

reg [31:0] stage_if_inst;
reg [31:0] stage_if_pc;
reg [31:0] stage_if_nextpc;

wire isbranch;
wire [31:0] branch_pc;

wire [31:0] if_pc_in;
assign if_pc_in = isbranch ? branch_pc : stage_if_nextpc;

wire [31:0] if_inst_out;
wire [31:0] if_nextpc_out;

wire id_stall_out;

Instfetch instfetch(
    if_pc_in,
    if_inst_out,
```

```

        if_if_pc_out
    );
// --- CLOCK ---
always @ (negedge clk_in)
begin
    if(reset)
    begin
        stage_if_inst <= 0;
        stage_if_pc <= 0;
        stage_if_nextpc <= 0;
    end
    else if (!id_stall_out)
    begin
        stage_if_inst <= if_inst_out;//当前指令 inst
        stage_if_pc <= if_pc_in;//当前指令 pc, 用于 ID
        stage_if_nextpc <= if_nextpc_out;//下一指令 pc, 如果没有跳转, 则赋给 if_pc_in
    end
end

reg [31:0] stage_id_alu_a;
reg [31:0] stage_id_alu_b;
reg [4:0] stage_id_aluchoice;
reg [1:0] stage_id_rfaddrinchoice;
reg [31:0] stage_id_cp0out;
reg [31:0] stage_id_hiout;
reg [31:0] stage_id_loout;
reg [2:0] stage_id_rfinchoice;
reg stage_id_rf_inallow;
reg stage_id_hi_inchoice;
reg stage_id_lo_inchoice;

reg [31:0] stage_id_inst;
reg [31:0] stage_id_rs;
reg [31:0] stage_id_rt;

reg [1:0] stage_id_dmem_inchoice;
reg [2:0] stage_id_dmem_outchoice;

//用于判断 stall
reg [31:0] stage_exe_inst;
reg [1:0] stage_exe_rfaddrinchoice;

//writeback 输入

```

```

reg [4:0]stage_id_wb_rfaddr;

wire [31:0] id_alu_a;
wire [31:0] id_alu_b;
wire [31:0] id_rs;
wire [31:0] id_rt;
wire [4:0] id_alu_choice;
wire [1:0] id_rf_addrinchoice;
wire [31:0] id_cp0out;
wire [31:0] id_hiout;
wire [31:0] id_loout;
wire [2:0] id_rf_inchoice;
wire id_rf_inallow;
wire id_hi_inchoice;
wire id_lo_inchoice;
wire [1:0]id_dmem_inchoice;
wire [2:0]id_dmem_outchoice;

//writeback 的输出，用于 id 的输入
wire [4:0]wb_rfaddr_out;
wire [31:0]wb_rf_out;
wire wb_rf_allow_out;
wire [31:0]wb_hi_out;
wire [31:0]wb_lo_out;

//writeback 输入
wire [4:0]id_wb_rfaddr;

//把 exe_out 和 mem_out 回接
wire [31:0]exe_alu_out;
wire [31:0]mem_wb_rf;

Instdecode instdecode(
    clk_in,
    reset,
    stage_if_pc,
    stage_if_inst,
    wb_rfaddr_out, //写 regfile
    wb_rf_out,
    wb_rf_allow_out,
    wb_hi_out, //写 hi_lo
    wb_lo_out,
    stage_id_inst, //判断 stall
    stage_id_rfaddrinchoice,

```

```

stage_exe_inst,
stage_exe_rfaddrinchoice,

//把 exe_out 和 mem_out 回接
exe_alu_out,
mem_wb_rf,

id_alu_a,
id_alu_b,
id_rs,
id_rt,
id_alu_choice, //入 alu 和 writeback
id_rf_addrinchoice,
id_rf_inchoice,
id_rf_inallow, //入 writeback
id_hi_inchoice,
id_lo_inchoice, //入 writeback

isbranch,
branch_pc,
id_stall_out,
id_cp0out,
id_hiout,
id_loout,
id_dmem_inchoice,
id_dmem_outchoice,
ans,

id_wb_rfaddr

);

// --- CLOCK ---
always @ (negedge clk_in)
begin
    if(reset)
        begin
            stage_id_alu_a <= 0;
            stage_id_alu_b <= 0;
            stage_id_aluchoice <= 0;
            stage_id_inst <= 0;
            stage_id_rs <= 0;
            stage_id_rt <= 0;
            stage_id_rfaddrinchoice <= 0;

```

```

stage_id_cp0out <= 0;
stage_id_hiout <= 0;
stage_id_loout <= 0;
stage_id_rfinchoice <= 0;
stage_id_rf_inallow <= 0;
stage_id_hi_inchoice <= 0;
stage_id_lo_inchoice <= 0;
stage_id_dmem_inchoice <= 0;
stage_id_dmem_outchoice <= 0;

//writeback 用
stage_id_wb_rfaddr <= 0;
end
else
begin
  if (!id_stall_out)
    begin
      stage_id_alu_a <= id_alu_a;
      stage_id_alu_b <= id_alu_b;
      stage_id_aluchoice <= id_alu_choice;

      //writeback 用
      stage_id_inst <= stage_if_inst;
      stage_id_rs <= id_rs;
      stage_id_rt <= id_rt;
      stage_id_rfaddrinchoice <= id_rf_addrinchoice;
      stage_id_cp0out <= id_cp0out;
      stage_id_hiout <= id_hiout;
      stage_id_loout <= id_loout;
      stage_id_rfinchoice <= id_rf_inchoice;
      stage_id_rf_inallow <= id_rf_inallow;
      stage_id_hi_inchoice <= id_hi_inchoice;
      stage_id_lo_inchoice <= id_lo_inchoice;
      stage_id_dmem_inchoice <= id_dmem_inchoice;
      stage_id_dmem_outchoice <= id_dmem_inchoice;

      //writeback 用
      stage_id_wb_rfaddr <= id_wb_rfaddr;
    end
  end
end

reg [31:0] stage_exe_alu_out;

```

```

wire [31:0] exe_alu_loout;

reg [31:0]stage_exe_rt;

reg [1:0]stage_exe_dmem_inchoice;
reg [2:0]stage_exe_dmem_outchoice;

reg [2:0] stage_exe_rfinchoice;
reg [31:0] stage_exe_cp0out;
reg [31:0] stage_exe_hiout;
reg [31:0] stage_exe_loout;
reg stage_exe_rf_inallow;

//writeback 输入
reg [4:0]stage_exe_wb_rfaddr;
reg [31:0]stage_exe_wb_hiin;
reg [31:0]stage_exe_wb_loin;
wire [31:0]exe_wb_hiin;
wire [31:0]exe_wb_loin;

Execute execute(
    stage_id_alu_a,
    stage_id_alu_b,
    stage_id_aluchoice,
    //writeback
    stage_id_hi_inchoice,
    stage_id_lo_inchoice,
    stage_id_rs,
    exe_alu_out,
    //wb
    exe_wb_hiin,
    exe_wb_loin
);

// --- CLOCK ---
always @ (negedge clk_in)
begin
    if(reset)
        begin
            stage_exe_alu_out <= 0;
            stage_exe_inst <= 0;

```

```

stage_exe_rfaddrinchoice <= 0;
stage_exe_cp0out <= 0;
stage_exe_hiout <= 0;
stage_exe_loout <= 0;
stage_exe_rfinchoice <= 0;
stage_exe_rf_inallow <= 0;

stage_exe_rt <= 0;
stage_exe_dmem_inchoice <= 0;
stage_exe_dmem_outchoice <= 0;

//writaback 用
stage_exe_wb_rfaddr <= 0;
stage_exe_wb_hiin <= 0;
stage_exe_wb_loin <= 0;
end
else
begin
    stage_exe_alu_out <= exe_alu_out;

    stage_exe_inst <= stage_id_inst;
    stage_exe_rfaddrinchoice <= stage_id_rfaddrinchoice;
    stage_exe_cp0out <= stage_id_cp0out;
    stage_exe_hiout <= stage_id_hiout;
    stage_exe_loout <= stage_id_loout;
    stage_exe_rfinchoice <= stage_id_rfinchoice;
    stage_exe_rf_inallow <= stage_id_rf_inallow;

    stage_exe_rt <= stage_id_rt;
    stage_exe_dmem_inchoice <= stage_id_dmem_inchoice;
    stage_exe_dmem_outchoice <= stage_id_dmem_outchoice;

    //writaback 用
    stage_exe_wb_rfaddr <= stage_id_wb_rfaddr;
    stage_exe_wb_hiin <= exe_wb_hiin;
    stage_exe_wb_loin <= exe_wb_loin;
end
end

//writaback 用
reg [4:0]stage_mem_wb_rfaddr;
reg [31:0]stage_mem_wb_rfin;
reg stage_mem_rf_inallow;
reg [31:0]stage_mem_wb_hiin;

```

```

reg [31:0]stage_mem_wb_loin;

Memory memory(
    clk_in,
    stage_exe_dmem_inchoice,
    stage_exe_alu_out,//也是 writeback 用
    stage_exe_rt,
    stage_exe_dmem_outchoice,
    stage_exe_rfchoice,//writeback 用
    stage_exe_wb_loin,
    stage_exe_cp0out,
    stage_exe_hiout,
    stage_exe_loout,
    mem_wb_rf
);

// --- CLOCK ---
always @ (negedge clk_in)
begin
    if(reset)
        begin
            stage_mem_wb_rfaddr <= 0;
            stage_mem_wb_rfin <= 0;
            stage_mem_rf_inallow <= 0;
            stage_mem_wb_hiin <= 0;
            stage_mem_wb_loin <= 0;
        end
    else
        begin
            stage_mem_wb_rfaddr <= stage_exe_wb_rfaddr;
            stage_mem_wb_rfin <= mem_wb_rf;
            stage_mem_rf_inallow <= stage_exe_rf_inallow;
            stage_mem_wb_hiin <= stage_exe_wb_hiin;
            stage_mem_wb_loin <= stage_exe_wb_loin;
        end
    end
end

Writeback writeback(
    stage_mem_wb_rfaddr,
    stage_mem_wb_rfin,
    stage_mem_rf_inallow,

```

```

    stage_mem_wb_hiin,
    stage_mem_wb_loin,

    wb_rfaddr_out,
    wb_rf_out,
    wb_rf_allow_out,
    wb_hi_out,
    wb_lo_out
);

seg7x16 seg(
    clk_in,
    reset,
    ans,
    o_seg,
    o_sel
);

endmodule

```

1.2 Instfetch

```

`timescale 1ns / 1ps
module Instfetch(
    input [31:0] if_pc_in,
    output [31:0] if_inst_out,
    output [31:0] if_nextpc_out
);
    imems cpu_imems(
        if_pc_in/4,
        if_inst_out
    );
//    imem cpu_imem
//    (
//        if_pc_in,
//        if_inst_out
//    );
    assign if_nextpc_out = if_pc_in + 4;
endmodule

```

1.3Instdecode

```
'timescale 1ns / 1ps
module Instdecode(
    input clk,
    input reset,
    input [31:0]id_pc_in,
    input [31:0]id_inst_in,
    input [4:0]id_rfaddr_in,
    input [31:0]id_rf_in,
    input id_rf_inallow_rf,
    input [31:0]id_hi_in,
    input [31:0]id_lo_in,

    input [31:0]stage_id_inst,///
    input [1:0]stage_id_rfaddrinchoice,
    input [31:0]stage_exe_inst,
    input [1:0]stage_exe_rfaddrinchoice,
    //传入 exe_out、 mem_out
    input [31:0]exe_out,
    input [31:0]mem_out,

    output [31:0]id_alu_a,
    output [31:0]id_alu_b,
    output [31:0]id_rs,
    output [31:0]id_rt,
    output [4:0]id_aluchoice,
    output [1:0]id_rf_addrinchoice,
    output [2:0]id_rf_inchoice,
    output id_rf_inallow,
    output id_hi_inchoice,
    output id_lo_inchoice,

    output isbranch,
    output [31:0]branch_pc,
    output id_stall_out,
    output [31:0]cp0_out,
    output [31:0]id_hi_out,
    output [31:0]id_lo_out, //来自 writeback

    output [1:0]id_dmem_inchoice,
    output [2:0]id_dmem_outchoice,
```

```

output [31:0]ans,
output [4:0]id_wb_rfaddr//wb
);
wire [1:0]alu_achoice;
wire [1:0]alu_bchoice;
wire mfc0src;
wire mtc0src;
wire exception;
wire _eret;
wire [4:0]cause;
wire [31:0]exc_addr;

wire [1:0] ischangea;
wire [1:0] ischangeb;

controlunit cpu_controlunit(
    id_inst_in,
    id_rs,
    id_rt,
    id_aluchoice,
    alu_achoice,
    alu_bchoice,
    id_rf_addrinchoice,
    id_rf_inchoice,
    id_rf_inallow,
    id_hi_inchoice,
    id_lo_inchoice,
    id_dmem_inchoice,
    id_dmem_outchoice,
    mfc0src,
    mtc0src,
    exception,
    _eret,
    cause
);

regfile cpu_ref (
    clk,
    reset,
    id_inst_in[25:21],
    id_inst_in[20:16],
    id_rfaddr_in,
    id_rf_in,

```

```

    id_rf_inallow_rf,
    ischangea,
    ischangeb,
    exe_out,
    mem_out,
    id_rs,
    id_rt,
    ans
);

cp0 cpu_cp0(
    clk,
    reset,
    mfc0src,
    mtc0src,
    id_pc_in,
    id_inst_in[15:11],
    id_rt,
    exception,
    _eret,
    cause,
    cp0_out,
    exc_addr
);

hi_lo cpu_hi_lo(
    clk,
    reset,
    id_hi_in,
    id_lo_in,
    id_hi_out,
    id_lo_out
);

branchpredict cpu_branchpredict(
    clk,
    id_pc_in,
    id_inst_in,
    id_stall_out,
    id_rs,
    id_alu_a,
    id_alu_b,
    isbranch,
    branch_pc
);

```

```

    );

    assign ischangea =  ((alu_achoice==2'b00 || alu_achoice==2'b01) &&
                        ((stage_id_rfaddrinchoice==2'b00
                        && stage_id_inst[15:11]!=0
                        && stage_id_inst[31:26]!=6'b101011) ||
                        (stage_id_rfaddrinchoice==2'b01
                        && stage_id_inst[20:16]!=0
                        && stage_id_inst[31:26]!=6'b101011)) ? 2'b01 :
                        ((alu_achoice==2'b00 || alu_achoice==2'b01) &&
                        ((stage_exe_rfaddrinchoice==2'b00
                        && stage_exe_inst[15:11]!=0
                        && stage_exe_inst[31:26]!=6'b101011) ||
                        (stage_exe_rfaddrinchoice==2'b01
                        && stage_exe_inst[20:16]!=0
                        && stage_exe_inst[31:26]!=6'b101011)) ? 2'b10 : 2'b00;

    assign ischangeb =  ((alu_bchoice==2'b00 || alu_bchoice==2'b10) &&
                        ((stage_id_rfaddrinchoice==2'b00
                        && stage_id_inst[15:11]!=0
                        && stage_id_inst[31:26]!=6'b101011) ||
                        (stage_id_rfaddrinchoice==2'b01
                        && stage_id_inst[20:16]==stage_id_inst[15:11]
                        && stage_id_inst[20:16]!=0
                        && stage_id_inst[31:26]!=6'b101011)) ? 2'b01 :
                        ((alu_bchoice==2'b00 || alu_bchoice==2'b10) &&
                        ((stage_exe_rfaddrinchoice==2'b00
                        && stage_exe_inst[15:11]!=0
                        && stage_exe_inst[31:26]!=6'b101011) ||
                        (stage_exe_rfaddrinchoice==2'b01
                        && stage_exe_inst[20:16]==stage_exe_inst[15:11]
                        && stage_exe_inst[20:16]!=0
                        && stage_exe_inst[31:26]!=6'b101011)) ? 2'b10 : 2'b00;

    assign id_alu_a=(alu_achoice==2'b00) ? id_rs :
                    (alu_achoice==2'b01) ? {{27{1'b0}},id_rs} :
                    (alu_achoice==2'b10) ? {{27{1'b0}},id_inst_in[10:6]} :
                    (alu_achoice==2'b11) ? id_rt : id_pc_in;

    assign id_alu_b=(alu_bchoice==2'b00) ? id_rt :
                    (alu_bchoice==2'b01) ? {{16{id_inst_in[15]}},id_inst_in[15:0]} :
                    (alu_bchoice==2'b10) ? {{16{1'b0}},id_inst_in[15:0]} : 32'b100;

    assign id_stall_out = 0;

    assign id_wb_rfaddr = id_rf_addrinchoice==2'b00 ? id_inst_in[15:11] :

```

```

        (id_rf_addrinchoice==2'b01 ? id_inst_in[20:16] : 5'b11111);
endmodule

```

1.4controlunit

```

`timescale 1ns / 1ps
module controlunit(
    input [31:0] inst,
    input [31:0] rs,
    input [31:0] rt,
    output [4:0]aluchoice,
    output [1:0]alu_achoice,
    output [1:0]alu_bchoice,
    output [1:0]rf_addrinchoice,
    output [2:0]rf_inchoice,
    output rf_inallow,
    output hi_inchoice,
    output lo_inchoice,
    output [1:0]dmem_inchoice,
    output [2:0]dmem_outchoice,
    output mfc0src,
    output mtc0src,
    output exception,
    output _eret,
    output [4:0]cause
);

wire [16:0]moreop;
assign moreop = {inst[31:21],inst[5:0]};
wire [11:0]opcode;
assign opcode = {inst[31:26],inst[5:0]};
wire [5:0]halfop;
assign halfop = {inst[31:26]};

parameter [5:0]
addi = 6'b001000,
addiu = 6'b001001,
andi = 6'b001100,
ori = 6'b001101,
sltiu = 6'b001011,
lui = 6'b001111,
xori = 6'b001110,

```

```
slti = 6'b0001010,  
beq = 12'b0000100,  
bne = 6'b0000101,  
bgez = 6'b0000001,  
  
j = 6'b0000010,  
jal = 6'b0000011,  
  
lw = 6'b100011,  
sw = 6'b101011,  
lb = 6'b100000,  
lbu = 6'b100100,  
lhu = 6'b100101,  
sb = 6'b101000,  
sh = 6'b101001,  
lh = 6'b100001;  
  
parameter [11:0]  
addu = 12'b000000_100001,  
_and = 12'b000000_100100,  
_xor = 12'b000000_100110,  
_nor = 12'b000000_100111,  
_or = 12'b000000_100101,  
sll = 12'b000000_000000,  
sllv = 12'b000000_000100,  
sltlu = 12'b000000_101011,  
sra = 12'b000000_000011,  
srl = 12'b000000_000010,  
subu = 12'b000000_100011,  
add = 12'b000000_100000,  
sub = 12'b000000_100010,  
slt = 12'b000000_101010,  
sriv = 12'b000000_000110,  
srav = 12'b000000_000111,  
clz = 12'b011100_100000,  
divu = 12'b000000_011011,  
mul = 12'b011100_000010,  
multu = 12'b000000_011001,  
teq = 12'b000000_110100,  
div = 12'b000000_011010,  
  
jr = 12'b000000_001000,  
jalr = 12'b000000_001001,
```

```

mfhi = 12'b000000_010000,
mflo = 12'b000000_010010,
mthi = 12'b000000_010001,
mtlo = 12'b000000_010011,

eret = 12'b010000_011000,
syscall = 12'b000000_001100,
_break = 12'b000000_001101;

parameter [16:0]
mfc0 = 17'b010000_00000_000000,
mtc0 = 17'b010000_00100_000000;

assign aluchoice =  (opcode==addu || halfop==addiu) ? 5'b00000:
                    (opcode==add || halfop==addi || halfop==lw || halfop==sw || halfop==lb || halfop==lbu || halfop==lhu || halfop==sb || halfop==sh || halfop==lh) ? 5'b00001:
                    (opcode==subu) ? 5'b00010:
                    (opcode==sub) ? 5'b00011:
                    (opcode==_and || halfop==andi) ? 5'b00100:
                    (opcode==_or || halfop==ori) ? 5'b00101:
                    (opcode==_xor || halfop==xori) ? 5'b00110:
                    (opcode==_nor) ? 5'b00111:
                    (halfop==lui) ? 5'b01000:
                    (opcode==sltu || halfop==sliu) ? 5'b01001:
                    (opcode==slt || halfop==sli) ? 5'b01010:
                    (opcode==sra || opcode==srav) ? 5'b01011:
                    (opcode==srl || opcode==sriv) ? 5'b01100:
                    (opcode==sll || opcode==sllv) ? 5'b01101:
                    (halfop==beq) ? 5'b01110:
                    (halfop==bne) ? 5'b01111:
                    (halfop==bgez) ? 5'b10000:
                    (opcode==div) ? 5'b10001:
                    (opcode==divu) ? 5'b10010:
                    (opcode==mul) ? 5'b10011:
                    (opcode==multu) ? 5'b10100:
                    (opcode==clz) ? 5'b10101:
                    (opcode==teq) ? 5'b10110: 5'b00000;

assign alu_achoice= (opcode==sll || opcode==srl || opcode==sra) ? 2'b10:
                    (opcode==sllv || opcode==sriv || opcode==srav) ? 2'b01: 2'b00;

assign alu_bchoice= (halfop==jal || halfop==jalr) ? 2'b11:
                    (halfop==andi || halfop==ori || halfop==xori || halfop==lw || halfop==sw ||
```

```

    halfop==lb || halfop==lbu || halfop==lhu || halfop==sb ||
halfop==sh || halfop==lh) ? 2'b10:
    (halfop==addi || halfop==addiu || halfop==sliu || halfop==lui ||
halfop==sli) ? 2'b01: 2'b00;

assign rf_addrinchoice= (halfop==addi || halfop==addiu || halfop==andi || halfop==ori ||
halfop==sliu || halfop==lui || halfop==xori || halfop==sli || halfop==lw || halfop==sw || halfop==lb || halfop==lbu || halfop==lhu || halfop==sb || halfop==sh || halfop==lh || moreop == mfc0) ? 2'b01:
(halfop==jal) ? 2'b10: 2'b00;

assign rf_inchoice= (halfop == jal || opcode == jalr) ? 3'b001:
(halfop == lw || halfop == lb || halfop == lbu || halfop == lhu || halfop == lh) ? 3'b010:
(opcode == mul || opcode == multu) ? 3'b011:
(moreop == mfc0) ? 3'b100:
(opcode == mfhi) ? 3'b101:
(opcode == mflo) ? 3'b110: 0;

assign rf_inallow = (halfop != sw && halfop != sh && halfop != sb && halfop != beq && halfop != bne) ? 1: 0;

assign hi_inchoice= (opcode == mthi) ? 1: 0;

assign lo_inchoice= (opcode == mtlo) ? 1: 0;

assign dmem_inchoice = (halfop==sw) ? 2'b01:
(halfop==sh) ? 2'b10:
(halfop==sb) ? 2'b11: 2'b00;

assign dmem_outchoice = (halfop==lh) ? 3'b001:
(halfop==lhu) ? 3'b010:
(halfop==lb) ? 3'b011:
(halfop==lbu) ? 3'b100: 3'b000;
//lw->dmem_outchoice==3'b000

assign exception = ((opcode==teq && rs==rt) || opcode==_break || opcode==syscall) ? 1: 0;

assign _eret = (opcode==eret) ? 1: 0;

assign cause = (opcode==teq && rs==rt) ? 5'b01101:
(opcode==_break) ? 5'b01001:
(opcode==syscall) ? 5'b01000:

```

```

        (opcode==_break) ? 5'b01001: 5'b00000;

assign mfc0src= (moreop == mfc0) ? 1: 0;

assign mtc0src= (moreop == mtc0) ? 1: 0;

endmodule

```

1.5regfile

```

`timescale 1ns / 1ps
module regfile(
    input clk,
    input rst,
    input [4:0] raddr1,
    input [4:0] raddr2, //所需读取的寄存器的地址
    input [4:0] waddr,
    input [31:0] wdata,
    input regfilesrc,//写寄存器的地址//写寄存器数据，数据在 clk 下降沿时被写入
    input [1:0]ischangea,
    input [1:0]ischangeb,
    input [31:0]exe_out,
    input [31:0]mem_out,
    output [31:0] rdata1,//raddr1 所对应寄存器的输出数据
    output [31:0] rdata2,//raddr2 所对应寄存器的输出数据
    output [31:0] ans
);
reg [31:0] array_reg [31:0]; //寄存器

assign rdata1 = (ischangea==2'b01) ? exe_out :
               (ischangea==2'b10) ? mem_out :
               (ischangea==2'b00 && raddr1==waddr && waddr!=0) ? wdata :
array_reg[raddr1];

assign rdata2 = (ischangeb==2'b01) ? exe_out :
               (ischangeb==2'b10) ? mem_out :
               (ischangeb==2'b00 && raddr2==waddr && waddr!=0) ? wdata :
array_reg[raddr2];

assign ans = array_reg[28];

//写寄存器
integer i;

```

```

always @(negedge clk)
begin
    if(rst)
        begin
            for(i=0;i<32;i=i+1)
                array_reg[i] <= 0;
        end
    else
        begin
            if((waddr!=0)&&regfilesrc==1)
                begin
                    array_reg[waddr] <= wdata;
                end
            else
                begin
                    array_reg[waddr] <= array_reg[waddr];
                end
        end
    end
endmodule

```

1.6cp0

```

module cp0(
    input clk,
    input rst,
    input mfc0,
    input mtc0,
    input [31:0]pc,
    input [4:0]Rd,//mfc0
    input [31:0]wdata,//mtc0
    input exception,
    input eret,
    input [4:0]cause,
    output [31:0]rdata,//mfc0
    output [31:0]exc_addr//pc
);
    integer i;

    reg[31:0] cp0[31:0];
    reg[31:0] status_temp;

    assign rdata = mfc0 ? cp0[Rd] : 32'hz;

```

```

assign exc_addr = (eret==1) ? cp0[14]:32'h00400004;

always@(negedge clk)
begin
    if(rst)
        begin
            for(i=0;i<32;i=i+1)
                cp0[i]<=0;
        end
    else
        begin
            if(mtc0)
                cp0[Rd] <= wdata;
            else
                if(exception)
                    begin
                        status_temp<=cp0[12];
                        if(eret==1'b0)
                            begin
                                cp0[12]<=cp0[12]<<5;
                                cp0[13]<={25'b0,cause,2'b0};
                                cp0[14]<=pc;
                            end
                        else
                            cp0[12]<=status_temp;
                    end
                end
        end
    end
endmodule

```

1.7hi_lo

```

`timescale 1ns / 1ps
module hi_lo(
    input clk,
    input rst,
    input [31:0] Write_hi,
    input [31:0] Write_lo,
    output [31:0] Rd_hi,
    output [31:0] Rd_lo
);
    reg [31:0] hi;
    reg [31:0] lo;

```

```

assign Rd_hi = hi;
assign Rd_lo = lo;

always @(negedge clk)
begin
    if(rst)
        begin
            hi <= 32'b0;
            lo <= 32'b0;
        end
    else
        begin
            hi <= Write_hi;
            lo <= Write_lo;
        end
    end
endmodule

```

1.8branchpredict

```

`timescale 1ns / 1ps
module branchpredict(
    input clk,
    input [31:0] pc,
    input [31:0] inst,
    input stall,
    input [31:0] rs,
    input [31:0] alu_a,
    input [31:0] alu_b,
    output isbranch,
    output [31:0]branch_pc
);

    wire [16:0]moreop;
    assign moreop = {inst[31:21],inst[5:0]};
    wire [11:0]opcode;
    assign opcode = {inst[31:26],inst[5:0]};
    wire [5:0]halfop;
    assign halfop = {inst[31:26]};

    parameter [5:0]
        beq = 12'b0000100,
        bne = 6'b000101,

```

```

bgez = 6'b000001,
j = 6'b000010,
jal = 6'b000011;

parameter [11:0]
jr = 12'b000000_001000,
jalr = 12'b000000_001001;

assign isbranch = (opcode==jr ||
                   opcode==jalr ||
                   halfop==j ||
                   halfop==jal ||
                   (halfop==beq && alu_a==alu_b) ||
                   (halfop==bne && alu_a!=alu_b) ||
                   (halfop==bgez && $signed(alu_a)>=0)) ? 1: 0;

assign branch_pc = (opcode==jr || opcode==jalr) ? rs-32'h00400000:
                           (halfop==j           ||           halfop==jal)      ?
{pc[31:28],inst[25:0]<<2>>-32'h00400000:
                           ((halfop==beq && alu_a==alu_b) ||
                            (halfop==bne && alu_a!=alu_b) ||
                            (halfop==bgez   &&   $signed(alu_a)>=0)) ? pc+{{(32 - 18){inst[15]}},inst[15:0],2'b00}+32'b100: 0;
endmodule

```

1.9 Execute

```

`timescale 1ns / 1ps
module Execute(
    input [31:0]exe_rs_in,
    input [31:0]exe_rt_in,
    input [4:0]exe_aluchoice,
    //writeback
    input exe_hi_inchoice,
    input exe_lo_inchoice,
    input [31:0]exe_rs,
    output [31:0]exe_alu_out,
    //writeback
    output [31:0]exe_wb_hiin,
    output [31:0]exe_wb_loin
)

```

```

);
wire [31:0]exe_hi_out;
wire [31:0]exe_lo_out;
alu cpu_alu(
    exe_rs_in,
    exe_rt_in,
    exe_aluchoice,
    exe_alu_out,
    exe_hi_out,
    exe_lo_out
);
assign exe_wb_hiin = exe_hi_inchoice ? exe_rs : exe_hi_out;
assign exe_wb_loin = exe_lo_inchoice ? exe_rs : exe_lo_out;
endmodule

```

1.10alu

```

`timescale 1ns / 1ps
module alu(
    input [31:0] A,
    input [31:0] B,
    input [4:0] aluchoice,
    output [31:0] result,
    output [31:0] hi_result,
    output [31:0] lo_result
);
wire [63:0] tomux;
wire [31:0] clz;

assign tomux = (aluchoice==5'b10011) ? $signed(A) * $signed(B):
               (aluchoice==5'b10100) ? A * B: 0;

assign clz= A[31]?0:A[30]?1:A[29]?2:A[28]?3:
           A[27]?4:A[26]?5:A[25]?6:A[24]?7:
           A[23]?8:A[22]?9:A[21]?10:A[20]?11:
           A[19]?12:A[18]?13:A[17]?14:A[16]?15:
           A[15]?16:A[14]?17:A[13]?18:A[12]?19:
           A[11]?20:A[10]?21:A[9]?22:A[8]?23:
           A[7]?24:A[6]?25:A[5]?26:A[4]?27:
           A[3]?28:A[2]?29:A[1]?30:A[0]?31:32;

```

```

assign result = (aluchoice==5'b00000) ? A + B://ADDU
    (aluchoice==5'b00001) ? $signed(A) + $signed(B)//ADD
    (aluchoice==5'b00010) ? A - B://SUBU
    (aluchoice==5'b00011) ? $signed(A) - $signed(B)//SUB
    (aluchoice==5'b00100) ? A & B://AND
    (aluchoice==5'b00101) ? A | B://OR
    (aluchoice==5'b00110) ? A ^ B://XOR
    (aluchoice==5'b00111) ? ~(A|B)//NOR
    (aluchoice==5'b01000) ? {B[15:0], 16'b0}//LUI
    (aluchoice==5'b01001) ? ((A < B) ? 1 : 0)//SLTU
    (aluchoice==5'b01010) ? (($signed(A)<$signed(B)) ? 1 : 0)//SLT
    (aluchoice==5'b01011) ? $signed(B) >> A://SRA 向右算术移位
    (aluchoice==5'b01100) ? B >> A://SRL 向右逻辑移位
    (aluchoice==5'b01101) ? B << A://SLL SLR
    (aluchoice==5'b10101) ? clz:
    (aluchoice==5'b10001) ? $signed(A) / $signed(B)//DIV
    (aluchoice==5'b10010) ? A / B://DIVU
    (aluchoice==5'b10011) ? tomux[31:0];//MUL
    (aluchoice==5'b10100) ? tomux[31:0]: 0;//MULTU

assign hi_result = (aluchoice==5'b10001) ? $signed(A) % $signed(B)//DIV
    (aluchoice==5'b10010) ? A % B://DIVU
    (aluchoice==5'b10011) ? tomux[63:32];//MUL
    (aluchoice==5'b10100) ? tomux[63:32]: 0;//MULTU

assign lo_result = (aluchoice==5'b10001) ? $signed(A) / $signed(B)//DIV
    (aluchoice==5'b10010) ? A / B://DIVU
    (aluchoice==5'b10011) ? tomux[31:0];//MUL
    (aluchoice==5'b10100) ? tomux[31:0]: 0;//MULTU

endmodule

```

1.11Memory

```

`timescale 1ns / 1ps
module Memory(
    input clk,
    input [1:0] mem_dmem_inchoice,
    input [31:0] mem_dmem_addr,//输入输出用一个地址
    input [31:0] mem_dmem_in,
    input [2:0] mem_dmem_outchoice,
    input [2:0]mem_rfinchoice,//writeback
    input [31:0]mem_alu_loout,
    input [31:0]mem_cp0out,

```

```

    input [31:0]mem_hiout,
    input [31:0]mem_loout,
    output [31:0] mem_wb_rf
);
wire [31:0]mem_dmem_out;

dmem cpu_dmem(
    clk,
    mem_dmem_inchoice,
    mem_dmem_addr,
    mem_dmem_in,
    mem_dmem_outchoice,
    mem_dmem_out
);

assign mem_wb_rf =  mem_rfinchoice==3'b000 ? mem_dmem_addr :
                    (mem_rfinchoice==3'b010 ? mem_dmem_out :
                     (mem_rfinchoice==3'b011 ? mem_alu_loout :
                      (mem_rfinchoice==3'b100 ? mem_cp0out :
                       (mem_rfinchoice==3'b101 ? mem_hiout : mem_loout))));

endmodule

```

1.12 dmem

```

`timescale 1ns / 1ps
module dmem(
    input clk,//时钟
    input [1:0]dmem_inchoice,
    input [31:0] addr,//地址
    input [31:0] data_in,//输入
    input [2:0]dmem_outchoice,
    output [31:0] data_out//输出
);

wire [7:0]Addr;
assign Addr = addr[7:0];
wire [7:0]lh_lb;
reg [7:0] num [0:1023];//寄存器
assign lh_lb = num[Addr];

assign      data_out      =          (dmem_outchoice==3'b000)      ?
{num[Addr],num[Addr+1],num[Addr+2],num[Addr+3]}:
(dmem_outchoice==3'b001)      ?

```

```

{{16{lh_lb[7]}},num[Addr],num[Addr+1]}:
    (dmem_outchoice==3'b010) ?
{{16{1'b0}},num[Addr],num[Addr+1]}:
    (dmem_outchoice==3'b011) ? {{24{lh_lb[7]}},num[Addr]}:
    (dmem_outchoice==3'b100) ? {{24{1'b0}},num[Addr]}: 0;

always@(negedge clk)
begin
    case (dmem_inchoice)
        2'b00:num[Addr]<=num[Addr];
        2'b01:
            begin
                num[Addr]<=data_in[31:24];
                num[Addr+1]<=data_in[23:16];
                num[Addr+2]<=data_in[15:8];
                num[Addr+3]<=data_in[7:0];
            end
        2'b10:
            begin
                num[Addr]<=data_in[15:8];
                num[Addr+1]<=data_in[7:0];
            end
        2'b11:num[Addr]<=data_in[7:0];
    endcase
end
endmodule

```

1.13 Writeback

```

`timescale 1ns / 1ps
module Writeback(
    input [4:0]wb_rfaddr_in,
    input [31:0]wb_rfin,
    input wb_rf_inallow,
    input [31:0]wb_hiin,
    input [31:0]wb_loin,
    output [4:0]wb_rfaddr_out,
    output [31:0]wb_rf_out,
    output wb_rf_allow,
    output [31:0]wb_hi_out,
    output [31:0]wb_lo_out
);
    assign wb_rfaddr_out = wb_rfaddr_in;

```

```

    assign wb_rf_out = wb_rfin;
    assign wb_rf_allow = wb_rf_inallow;
    assign wb_hi_out = wb_hiin;
    assign wb_lo_out = wb_loin;
endmodule

```

1.14seg7x16

```

`timescale 1ns / 1ps
module seg7x16(
    input clk,
    input reset,
    input [31:0] i_data,
    output [7:0] o_seg,
    output [7:0] o_sel
);

// parameter period= 10000;
// reg [31:0]cnt;
// reg clk_out;
// always @(posedge clk, posedge reset)//分频 500Hz
// begin
//     if(reset)
//         cnt <= 0 ;
//     else
//     begin
//         cnt<= cnt+1;
//         if(cnt== (period >> 1) - 1)
//             clk_out <= 1'b1;
//         else if(cnt == period - 1)
//             begin
//                 clk_out <= 1'b0;
//                 cnt <= 1'b0;
//             end
//         end
//     end
// end

// wire seg7_clk = clk_out;
// wire seg7_clk = clk;
reg [2:0] seg7_addr;

always @ (posedge seg7_clk, posedge reset)
if(reset)

```

```

    seg7_addr <= 0;
else
    seg7_addr <= seg7_addr + 1'b1;

reg [7:0] o_sel_r;
always @ (*)
case(seg7_addr)
    7 : o_sel_r = 8'b01111111;
    6 : o_sel_r = 8'b10111111;
    5 : o_sel_r = 8'b11011111;
    4 : o_sel_r = 8'b11101111;
    3 : o_sel_r = 8'b11110111;
    2 : o_sel_r = 8'b11111011;
    1 : o_sel_r = 8'b11111101;
    0 : o_sel_r = 8'b11111110;
endcase

reg [31:0] i_data_store;
always @ (posedge clk, posedge reset)
if(reset)
    i_data_store <= 0;
else
    i_data_store <= i_data;

reg [7:0] seg_data_r;
always @ (*)
case(seg7_addr)
    0 : seg_data_r = i_data_store[3:0];
    1 : seg_data_r = i_data_store[7:4];
    2 : seg_data_r = i_data_store[11:8];
    3 : seg_data_r = i_data_store[15:12];
    4 : seg_data_r = i_data_store[19:16];
    5 : seg_data_r = i_data_store[23:20];
    6 : seg_data_r = i_data_store[27:24];
    7 : seg_data_r = i_data_store[31:28];
endcase

reg [7:0] o_seg_r;
always @ (posedge clk, posedge reset)
if(reset)
    o_seg_r <= 8'hff;
else
    case(seg_data_r)
        4'h0 : o_seg_r <= 8'hC0;

```

```

4'h1 : o_seg_r <= 8'hF9;
4'h2 : o_seg_r <= 8'hA4;
4'h3 : o_seg_r <= 8'hB0;
4'h4 : o_seg_r <= 8'h99;
4'h5 : o_seg_r <= 8'h92;
4'h6 : o_seg_r <= 8'h82;
4'h7 : o_seg_r <= 8'hF8;
4'h8 : o_seg_r <= 8'h80;
4'h9 : o_seg_r <= 8'h90;
4'hA : o_seg_r <= 8'h88;
4'hB : o_seg_r <= 8'h83;
4'hC : o_seg_r <= 8'hC6;
4'hD : o_seg_r <= 8'hA1;
4'hE : o_seg_r <= 8'h86;
4'hF : o_seg_r <= 8'h8E;
endcase

assign o_sel = o_sel_r;
assign o_seg = o_seg_r;

endmodule

```

1.15clk_wiz

```

`timescale 1ns / 1ps
module clk_wiz(
    input clk_in,
    input reset,
    output the_clk_out
);
//    parameter period= 20;
parameter period = 400000;
reg [31:0]cnt;
reg clk_out;
always @(negedge clk_in)//分频
begin
    if(reset)
        begin
            cnt <= 0;
            clk_out <= 0;
        end
    else
        begin

```

```

if(cnt== (period >> 1) - 1)
begin
    clk_out <= 1'b1;
    cnt<= cnt+1;
end
else if(cnt == period - 1)
begin
    clk_out <= 1'b0;
    cnt <= 1'b0;
end
else
begin
    cnt<= cnt+1;
end
end
end

assign the_clk_out = clk_out;
endmodule

```

2、静态流水线的设计程序

2.1sccomp_dataflow

```

`timescale 1ns / 1ps
module sccomp_dataflow(
    input clk_in, //时钟
    input reset, //复位
    output [7:0] o_seg,
    output [7:0] o_sel
);
wire [31:0]ans;
//wire clk_in;
//clk_wiz Myclk(clk, clk_in);

reg [31:0] stage_if_inst;
reg [31:0] stage_if_pc;
reg [31:0] stage_if_nextpc;

wire isbranch;
wire [31:0]branch_pc;

wire [31:0] if_pc_in = isbranch ? branch_pc : stage_if_nextpc;

```

```

wire [31:0] if_inst_out;
wire [31:0] if_nextpc_out;

wire id_stall_out;

InstFetch instfetch(
    if_pc_in,
    if_inst_out,
    if_nextpc_out
);
// --- CLOCK ---
always @ (negedge clk_in)
begin
    if(reset)
        begin
            stage_if_inst <= 0;
            stage_if_pc <= 0;
            stage_if_nextpc <= 0;
        end
    else if (!id_stall_out)
        begin
            stage_if_inst <= if_inst_out;//当前指令 inst
            stage_if_pc <= if_pc_in;//当前指令 pc, 用于 ID
            stage_if_nextpc <= if_nextpc_out;//下一指令 pc, 如果没有跳转, 则赋给 if_pc_in
        end
end
end

reg [31:0] stage_id_alu_a;
reg [31:0] stage_id_alu_b;
reg [4:0] stage_id_aluchoice;
reg [1:0] stage_id_rfaddrinchoice;
reg [31:0] stage_id_cp0out;
reg [31:0] stage_id_hiout;
reg [31:0] stage_id_loout;
reg [2:0] stage_id_rfchoice;
reg stage_id_rf_inallow;
reg stage_id_hi_inchoice;
reg stage_id_lo_inchoice;

wire [31:0] id_alu_a;
wire [31:0] id_alu_b;

```

```

wire [31:0] id_rs;
wire [31:0] id_rt;
wire [4:0] id_alu_choice;
wire [1:0] id_rf_addrinchoice;
wire [31:0] id_cp0out;
wire [31:0] id_hiout;
wire [31:0] id_loout;
wire [2:0] id_rf_inchoice;
wire id_rf_inallow;
wire id_hi_inchoice;
wire id_lo_inchoice;
wire [1:0]id_dmem_inchoice;
wire [2:0]id_dmem_outchoice;

reg [31:0] stage_id_inst;
reg [31:0] stage_id_rs;
reg [31:0] stage_id_rt;

reg [1:0] stage_id_dmem_inchoice;
reg [2:0] stage_id_dmem_outchoice;

//用于判断 stall
reg [31:0]stage_exe_inst;
reg [1:0] stage_exe_rfaddrinchoice;

//writeback 的输出，用于 id 的输入
wire [4:0]wb_rfaddr_out;
wire [31:0]wb_rf_out;
wire wb_rf_allow;
wire [31:0]wb_hi_out;
wire [31:0]wb_lo_out;

InstDecode instdecode(
    clk_in, reset,
    stage_if_pc, stage_if_inst,
    wb_rfaddr_out, wb_rf_out, wb_rf_allow, //写 regfile
    wb_hi_out, wb_lo_out, //写 hi_lo
    stage_id_inst, stage_id_rfaddrinchoice,//判断 stall
    stage_exe_inst, stage_exe_rfaddrinchoice,

    id_alu_a, id_alu_b, id_rs, id_rt, id_alu_choice, //入 alu 和 writeback
    id_rf_addrinchoice, id_rf_inchoice, id_rf_inallow,//入 writeback
    id_hi_inchoice, id_lo_inchoice,//入 writeback

```

```

    isbranch, branch_pc,
    id_stall_out,
    id_cp0out, id_hiout, id_loout,
    id_dmem_inchoice, id_dmem_outchoice,
    ans
);

// --- CLOCK ---
always @ (negedge clk_in)
begin
    if(reset)
        begin
            stage_id_alu_a <= 0;
            stage_id_alu_b <= 0;
            stage_id_aluchoice <= 0;
            stage_id_inst <= 0;
            stage_id_rs <= 0;
            stage_id_rt <= 0;
            stage_id_rfaddrinchoice <= 0;
            stage_id_cp0out <= 0;
            stage_id_hiout <= 0;
            stage_id_loout <= 0;
            stage_id_rf_inallow <= 0;
            stage_id_hi_inchoice <= 0;
            stage_id_lo_inchoice <= 0;
            stage_id_dmem_inchoice <= 0;
            stage_id_dmem_outchoice <= 0;
        end
    else
        begin
            if (!id_stall_out)
                begin
                    stage_id_alu_a <= id_alu_a;
                    stage_id_alu_b <= id_alu_b;
                    stage_id_aluchoice <= id_alu_choice;

                    //writeback 用
                    stage_id_inst <= stage_if_inst;
                    stage_id_rs <= id_rs;
                    stage_id_rt <= id_rt;
                    stage_id_rfaddrinchoice <= id_rf_addrinchoice;
                    stage_id_cp0out <= id_cp0out;
                    stage_id_hiout <= id_hiout;
                end
        end
end

```

```

        stage_id_loout <= id_loout;
        stage_id_rfinchoice <= id_rf_inchoice;
        stage_id_rf_inallow <= id_rf_inallow;
        stage_id_hi_inchoice <= id_hi_inchoice;
        stage_id_lo_inchoice <= id_lo_inchoice;
        stage_id_dmem_inchoice <= id_dmem_inchoice;
        stage_id_dmem_outchoice <= id_dmem_inchoice;
    end
end
end

reg [31:0] stage_exe_alu_out;
reg [31:0] stage_exe_alu_hiout;
reg [31:0] stage_exe_alu_loout;

wire [31:0] exe_alu_out;
wire [31:0] exe_alu_hiout;
wire [31:0] exe_alu_loout;

reg [31:0]stage_exe_rs;
reg [31:0]stage_exe_rt;

reg [1:0]stage_exe_dmem_inchoice;
reg [2:0]stage_exe_dmem_outchoice;

reg [2:0] stage_exe_rfinchoice;
reg [31:0] stage_exe_cp0out;
reg [31:0] stage_exe_hiout;
reg [31:0] stage_exe_loout;
reg stage_exe_rf_inallow;
reg stage_exe_hi_inchoice;
reg stage_exe_lo_inchoice;

Execute execute(
    clk_in,
    stage_id_alu_a,
    stage_id_alu_b,
    stage_id_aluchoice,
    exe_alu_out,
    exe_alu_hiout,
    exe_alu_loout
);

```

```

// --- CLOCK ---
always @ (negedge clk_in)
begin
    if(reset)
        begin
            stage_exe_alu_out <= 0;
            stage_exe_alu_hiout <= 0;
            stage_exe_alu_loout <= 0;
            stage_exe_inst <= 0;
            stage_exe_rs <= 0;
            stage_exe_rfaddrinchoice <= 0;
            stage_exe_cp0out <= 0;
            stage_exe_hiout <= 0;
            stage_exe_loout <= 0;
            stage_exe_rfinchoice <= 0;
            stage_exe_rf_inallow <= 0;
            stage_exe_hi_inchoice <= 0;
            stage_exe_lo_inchoice <= 0;

            stage_exe_rt <= 0;
            stage_exe_dmem_inchoice <= 0;
            stage_exe_dmem_outchoice <= 0;
        end
    else
        begin
            stage_exe_alu_out <= exe_alu_out;// 
            stage_exe_alu_hiout <= exe_alu_hiout;// 
            stage_exe_alu_loout <= exe_alu_loout;// 

            stage_exe_inst <= stage_id_inst;
            stage_exe_rs <= stage_id_rs;
            stage_exe_rfaddrinchoice <= stage_id_rfaddrinchoice;
            stage_exe_cp0out <= stage_id_cp0out;
            stage_exe_hiout <= stage_id_hiout;
            stage_exe_loout <= stage_id_loout;
            stage_exe_rfinchoice <= stage_id_rfinchoice;
            stage_exe_rf_inallow <= stage_id_rf_inallow;
            stage_exe_hi_inchoice <= stage_id_hi_inchoice;
            stage_exe_lo_inchoice <= stage_id_lo_inchoice;

            stage_exe_rt <= stage_id_rt;
            stage_exe_dmem_inchoice <= stage_id_dmem_inchoice;
            stage_exe_dmem_outchoice <= stage_id_dmem_outchoice;
        end
    end
end

```

```

        end
    end

reg [31:0]stage_mem_dmem_out;

wire [31:0]mem_dmem_out;

reg [31:0] stage_mem_alu_out;
reg [31:0] stage_mem_alu_hiout;
reg [31:0] stage_mem_alu_loout;
reg [31:0]stage_mem_inst;
reg [31:0]stage_mem_rs;
reg [1:0] stage_mem_rfaddrinchoice;
reg [31:0] stage_mem_cp0out;
reg [31:0] stage_mem_hiout;
reg [31:0] stage_mem_loout;
reg [2:0] stage_mem_rfinchoice;
reg stage_mem_rf_inallow;
reg stage_mem_hi_inchoice;
reg stage_mem_lo_inchoice;

```

```

Memory memory(
    clk_in,
    stage_exe_dmem_inchoice,
    stage_exe_alu_out,
    stage_exe_rt,
    stage_exe_dmem_outchoice,
    mem_dmem_out
);

```

```

// --- CLOCK ---
always @ (negedge clk_in)
begin
    if(reset)
        begin
            stage_mem_dmem_out <= 0;
            stage_mem_alu_out <= 0;
            stage_mem_alu_hiout <= 0;
            stage_mem_alu_loout <= 0;
            stage_mem_inst <= 0;
            stage_mem_rs <= 0;
            stage_mem_rfaddrinchoice <= 0;
            stage_mem_rfinchoice <= 0;

```

```

    stage_mem_rf_inallow <= 0;
    stage_mem_hi_inchoice <= 0;
    stage_mem_lo_inchoice <= 0;
end
else
begin
    stage_mem_dmem_out <= mem_dmem_out;

    stage_mem_alu_out <= stage_exe_alu_out;
    stage_mem_alu_hiout <= stage_exe_alu_hiout;
    stage_mem_alu_loout <= stage_exe_alu_loout;
    stage_mem_inst <= stage_exe_inst;
    stage_mem_rs <= stage_exe_rs;
    stage_mem_rfaddrinchoice <= stage_exe_rfaddrinchoice;
    stage_mem_cp0out <= stage_exe_cp0out;
    stage_mem_hiout <= stage_exe_hiout;
    stage_mem_loout <= stage_exe_loout;
    stage_mem_rfinchoice <= stage_exe_rfinchoice;
    stage_mem_rf_inallow <= stage_exe_rf_inallow;
    stage_mem_hi_inchoice <= stage_exe_hi_inchoice;
    stage_mem_lo_inchoice <= stage_exe_lo_inchoice;
end
end

```

Writeback writeback(

```

    stage_mem_rfaddrinchoice,
    stage_mem_inst[15:11],
    stage_mem_inst[20:16],
    5'b11111,
```

```

    stage_mem_rfinchoice,
    stage_mem_alu_out,
    stage_mem_dmem_out,
    stage_mem_alu_loout,
    stage_mem_cp0out,
    stage_mem_hiout,
    stage_mem_loout,
```

```
    stage_mem_rf_inallow,
```

```

    stage_mem_hi_inchoice,
    stage_mem_rs,
```

```

    stage_mem_alu_hiout,
    stage_mem_lo_inchoice,
    stage_mem_rs,
    stage_mem_alu_loout,
    wb_rfaddr_out,
    wb_rf_out,
    wb_rf_allow,
    wb_hi_out,
    wb_lo_out
);
seg7x16 Seg(
    clk_in,
    reset,
    32'h9fd0b7ea,
    o_seg,
    o_sel
);
endmodule

```

2.2 InstFetch

```

`timescale 1ns / 1ps
module InstFetch(
    input [31:0] if_pc_in,
    output [31:0] if_inst_out,
    output [31:0] if_nextpc_out
);
//    imem cpu_imem
//    (
//        if_pc_in,
//        if_inst_out
//    );
    imems cpu_imems(
        if_pc_in/4,
        if_inst_out
    );
    assign if_nextpc_out = if_pc_in + 4;
endmodule

```

2.3InstDecode

```
'timescale 1ns / 1ps
module InstDecode(
    input clk,
    input reset,
    input [31:0]id_pc_in,
    input [31:0]id_inst_in,
    input [4:0]id_rfaddr_in,
    input [31:0]id_rf_in,
    input id_rf_inallow_rf,
    input [31:0]id_hi_in,
    input [31:0]id_lo_in,

    input [31:0]stage_id_inst,
    input [1:0]stage_id_rfaddrinchoice,
    input [31:0]stage_exe_inst,
    input [1:0]stage_exe_rfaddrinchoice,

    output [31:0]id_alu_a,
    output [31:0]id_alu_b,
    output [31:0]id_rs,
    output [31:0]id_rt,
    output [4:0]id_aluchoice,
    output [1:0]id_rf_addrinchoice,
    output [2:0]id_rf_inchoice,
    output id_rf_inallow,
    output id_hi_inchoice,
    output id_lo_inchoice,

    output isbranch,
    output [31:0]branch_pc,
    output id_stall_out,
    output [31:0]cp0_out,
    output [31:0]id_hi_out,
    output [31:0]id_lo_out, //要用到 rf_in 里，在 writeback

    output [1:0]id_dmem_inchoice,
    output [2:0]id_dmem_outchoice,
    output [31:0]ans
);

wire [1:0]alu_achoice;
```

```

wire [1:0]alu_bchoice;

wire mfc0src;
wire mtc0src;
wire exception;
wire _eret;
wire [4:0]cause;
wire [31:0]exc_addr;

controlunit cpu_controlunit(
    clk,
    id_inst_in,
    id_rs, id_rt,
    id_aluchoice, alu_achoice, alu_bchoice,
    id_rf_addrinchoice, id_rf_inchoice, id_rf_inallow,
    id_hi_inchoice, id_lo_inchoice,
    id_dmem_inchoice, id_dmem_outchoice,
    mfc0src, mtc0src, exception, _eret, cause
);

regfile cpu_ref (
    clk, reset,
    id_inst_in[25:21], id_inst_in[20:16],
    id_rfaddr_in, id_rf_in, id_rf_inallow_rf,
    id_rs, id_rt,
    ans
);

cp0 cpu_cp0(
    clk,
    reset,
    mfc0src,
    mtc0src,
    id_pc_in,
    id_inst_in[15:11],
    id_rt,
    exception,
    _eret,
    cause,
    cp0_out,
    exc_addr);

```

```

hi_lo cpu_hi_lo(clk, reset, id_hi_in, id_lo_in, id_hi_out, id_lo_out);

assign id_alu_a = alu_achoice==2'b00 ? id_rs :
    (alu_achoice==2'b01 ? {{27{1'b0}},id_rs} :
     (alu_achoice==2'b10 ? {{27{1'b0}},id_inst_in[10:6]} : id_pc_in));
assign id_alu_b = alu_bchoice==2'b00 ? id_rt :
    (alu_bchoice==2'b01 ? {{16{id_inst_in[15]}},id_inst_in[15:0]} :
     (alu_bchoice==2'b10 ? {{16{1'b0}},id_inst_in[15:0]} : 32'b100));

//stall
reg [1:0]stall_count;
always @(posedge clk)
begin
    if(!id_stall_out && stage_id_inst)
        begin
            if(({stage_id_inst[31:26],stage_id_inst[5:0]}==12'b000000_011011 ||
                {stage_id_inst[31:26],stage_id_inst[5:0]}==12'b000000_011010 ||
                {stage_id_inst[31:26],stage_id_inst[5:0]}==12'b011100_000010 ||
                {stage_id_inst[31:26],stage_id_inst[5:0]}==12'b000000_011001) &&
                ({id_inst_in[31:26],id_inst_in[5:0]}==12'b000000_010000 ||
                 {id_inst_in[31:26],id_inst_in[5:0]}==12'b000000_010010))
            begin
                stall_count <= 2'b11;
            end
            else if(({stage_exe_inst[31:26],stage_exe_inst[5:0]}==12'b000000_011011 ||
                     {stage_exe_inst[31:26],stage_exe_inst[5:0]}==12'b000000_011010 ||
                     {stage_exe_inst[31:26],stage_exe_inst[5:0]}==12'b011100_000010 ||
                     {stage_exe_inst[31:26],stage_exe_inst[5:0]}==12'b000000_011001) &&
                     ({id_inst_in[31:26],id_inst_in[5:0]}==12'b000000_010000 ||
                      {id_inst_in[31:26],id_inst_in[5:0]}==12'b000000_010010))
            begin
                stall_count <= 2'b10;
            end
            if(alu_achoice==2'b00 || alu_achoice==2'b01)
                begin
                    if(stage_id_rfaddrinchoice==2'b00 && id_inst_in[25:21]==stage_id_inst[15:11]
                    && stage_id_inst[15:11]!=0)
                    begin
                        stall_count <= 2'b11;
                    end
                    else if(stage_exe_rfaddrinchoice==2'b00

```

```

id_inst_in[25:21]==stage_exe_inst[15:11] && stage_exe_inst[15:11]!=0
begin
    stall_count <= 2'b10;
end
else if(stage_id_rfaddrinchoice==2'b01) &&
id_inst_in[25:21]==stage_id_inst[20:16] && stage_id_inst[20:16]!=0
begin
    stall_count <= 2'b11;
end
else if(stage_exe_rfaddrinchoice==2'b01) &&
id_inst_in[25:21]==stage_exe_inst[20:16] && stage_exe_inst[20:16]!=0
begin
    stall_count <= 2'b10;
end
end
if(alu_bchoice==2'b00 && stall_count!=2'b11)
begin
    if(stage_id_rfaddrinchoice==2'b00 && id_inst_in[20:16]==stage_id_inst[15:11]
&& stage_id_inst[15:11]!=0)
begin
    stall_count <= 2'b11;
end
else if(stage_exe_rfaddrinchoice==2'b00) &&
id_inst_in[20:16]==stage_exe_inst[15:11] && stage_exe_inst[15:11]!=0
begin
    stall_count <= 2'b10;
end
else if(stage_id_rfaddrinchoice==2'b01) &&
id_inst_in[20:16]==stage_id_inst[20:16] && stage_id_inst[20:16]!=0
begin
    stall_count <= 2'b11;
end
else if(stage_exe_rfaddrinchoice==2'b01) &&
id_inst_in[20:16]==stage_exe_inst[20:16] && stage_exe_inst[20:16]!=0
begin
    stall_count <= 2'b10;
end
end
end
if(reset)
begin
    stall_count <= 2'b00;
end
else

```

```

begin
    if(stall_count!=2'b00)
        begin
            stall_count <= stall_count-1;
        end
    end
end

assign id_stall_out = stall_count ? 1 : 0;

//isbranch
branchpredict cpu_branchpredict(
    clk,
    id_pc_in, id_inst_in,
    id_stall_out,
    id_rs, id_alu_a, id_alu_b,
    isbranch, branch_pc
);

endmodule

```

2.4controlunit

```

`timescale 1ns / 1ps
module controlunit(
    input clk,
    input [31:0] inst,
    input [31:0] rs,
    input [31:0] rt,
    output reg [4:0]aluchoice,
    output reg [1:0]alu_achoice,
    output reg [1:0]alu_bchoice,
    output reg [1:0]rf_addrinchoice,
    output reg [2:0]rf_inchoice,
    output reg rf_inallow,
    output reg hi_inchoice,
    output reg lo_inchoice,
    output reg [1:0]dmem_inchoice,
    output reg [2:0]dmem_outchoice,

    output reg mfc0src,
    output reg mtc0src,
    output reg exception,

```

```

output reg _eret,
output reg [4:0]cause
);
wire [16:0]moreop;
assign moreop = {inst[31:21],inst[5:0]};
wire [11:0]opcode;
assign opcode = {inst[31:26],inst[5:0]};
wire [5:0]halfop;
assign halfop = {inst[31:26]};

parameter [5:0]
addi = 6'b001000,
addiu = 6'b001001,
andi = 6'b001100,
ori = 6'b001101,
sltiu = 6'b001011,
lui = 6'b001111,
xori = 6'b001110,
slti = 6'b001010,
beq = 12'b0000100,
bne = 6'b0000101,
bgez = 6'b000001,

j = 6'b0000010,
jal = 6'b0000011,

lw = 6'b100011,
sw = 6'b101011,
lb = 6'b100000,
lbu = 6'b100100,
lhu = 6'b100101,
sb = 6'b101000,
sh = 6'b101001,
lh = 6'b100001;

parameter [11:0]
addu = 12'b000000_100001,
_and = 12'b000000_100100,
_xor = 12'b000000_100110,
_nor = 12'b000000_100111,
_or = 12'b000000_100101,
sll = 12'b000000_000000,
sllv = 12'b000000_000100,
sltlu = 12'b000000_101011,

```

```

sra = 12'b000000_000011,
srl = 12'b000000_000010,
subu = 12'b000000_100011,
add = 12'b000000_100000,
sub = 12'b000000_100010,
slt = 12'b000000_101010,
srlv = 12'b000000_000110,
srav = 12'b000000_000111,
clz = 12'b011100_100000,
divu = 12'b000000_011011,
mul = 12'b011100_000010,
multu = 12'b000000_011001,
teq = 12'b000000_110100,
div = 12'b000000_011010,

jr = 12'b000000_001000,
jalr = 12'b000000_001001,

mfhi = 12'b000000_010000,
mflo = 12'b000000_010010,
mthi = 12'b000000_010001,
mtlo = 12'b000000_010011,

eret = 12'b010000_011000,
syscall = 12'b000000_001100,
_break = 12'b000000_001101;

parameter [16:0]
mfc0 = 17'b010000_00000_000000,
mtc0 = 17'b010000_00100_000000;

//根据 opcode 和当前状态确认当前控制信号
always @(posedge clk)
begin;
    //alu 的操作
    case(opcode)
        addu,addiu:aluchoice <= 5'b00000;
        add,addi:aluchoice <= 5'b00001;
        subu:aluchoice <= 5'b00010;
        sub:aluchoice <= 5'b00011;
        andi,_and:aluchoice <= 5'b00100;
        ori,_or:aluchoice <= 5'b00101;
        xor,_xor:aluchoice <= 5'b00110;
        _nor:aluchoice <= 5'b00111;
    endcase
end

```

```

lui:aluchoice <= 5'b01000;
slt,slt:aluchoice <= 5'b01001;
slt,slt:aluchoice <= 5'b01010;
sra,srav:aluchoice <= 5'b01011;
srl,srlv:aluchoice <= 5'b01100;
sll,sllv:aluchoice <= 5'b01101;
beq:aluchoice <= 5'b01110;
bne:aluchoice <= 5'b01111;
bgez:aluchoice <= 5'b10000;
div:aluchoice <= 5'b10001;
divu:aluchoice <= 5'b10010;
mul:aluchoice <= 5'b10011;
multu:aluchoice <= 5'b10100;
clz:aluchoice <= 5'b10101;
teq:aluchoice <= 5'b10110;
default: aluchoice <= 5'b00000;
endcase
case(halfop)
    addiu:aluchoice <= 5'b00000;
    addi, lw, sw, lb, lbu, lhu, sb, sh, lh:aluchoice <= 5'b00001;
    andi:aluchoice <= 5'b00100;
    ori:aluchoice <= 5'b00101;
    xori:aluchoice <= 5'b00110;
    lui:aluchoice <= 5'b01000;
    sltiu:aluchoice <= 5'b01001;
    slti:aluchoice <= 5'b01010;
    beq:aluchoice <= 5'b01110;
    bne:aluchoice <= 5'b01111;
    bgez:aluchoice <= 5'b10000;
    default: aluchoice <= aluchoice;
endcase
//alu_a 选择
case(opcode)
    sll, srl, sra:alu_achoice <= 2'b10;
    sllv, srlv, srav:alu_achoice <= 2'b01;
    default: alu_achoice <= 2'b00;
endcase
//alu_b 选择
case(halfop)
    jal, jalr:alu_bchoice <= 2'b11;
    andi, ori, xori, lw, sw, lb, lbu, lhu, sb, sh, lh:alu_bchoice <= 2'b10;
    addi,addiu, sltiu, lui, slti:alu_bchoice <= 2'b01;
    default: alu_bchoice <= 2'b00;
endcase

```

```

//rf_addrin 选择
case(halfop)
    addi,addiu, andi, ori, sltiu, lui, xori, slti, lw, sw, lb, lbu, lhu, sb, sh,
lh:rf_addrinchoice <= 2'b01;
    jal:rf_addrinchoice <= 2'b10;
    default: rf_addrinchoice <= 2'b00;
endcase
if(moreop == mfc0)
    rf_addrinchoice <= 2'b01;
else
    rf_addrinchoice <= rf_addrinchoice;
//rf_in 选择
if(halfop == jal || opcode == jalr)
    rf_inchoice <= 3'b001;
else if(halfop == lw || halfop == lb || halfop == lbu || halfop == lhu || halfop == lh)
    rf_inchoice <= 3'b010;
else if(opcode == mul || opcode == multu)
    rf_inchoice <= 3'b011;
else if(moreop == mfc0)
    rf_inchoice <= 3'b100;
else if(opcode == mfhi)
    rf_inchoice <= 3'b101;
else if(opcode == mflo)
    rf_inchoice <= 3'b110;
else
    rf_inchoice <= 0;
//rf_inallow 选择
if (halfop != sw && halfop != sh && halfop != sb && halfop != beq && halfop != bne)
    rf_inallow <= 1;
else
    rf_inallow <= 0;
//hi_lo
if(opcode == mthi)
begin
    hi_inchoice <= 1;
    lo_inchoice <= 0;
end
else if(opcode == mtlo)
begin
    hi_inchoice <= 0;
    lo_inchoice <= 1;
end
else
begin

```

```

    hi_inchoice <= 0;
    lo_inchoice <= 0;
end
//dmem
case(halfop)
    sw:dmem_inchoice <= 2'b01;
    sh:dmem_inchoice <= 2'b10;
    sb:dmem_inchoice <= 2'b11;
    default: dmem_inchoice <= 2'b00;
endcase
case(halfop)
    lw:dmem_outchoice <= 3'b000;
    lh:dmem_outchoice <= 3'b001;
    lhu:dmem_outchoice <= 3'b010;
    lb:dmem_outchoice <= 3'b011;
    lbu:dmem_outchoice <= 3'b100;
    default: dmem_outchoice <= 3'b000;
endcase
//cp0
case(opcode)
    teq:
begin
    if(rs == rt)
begin
        exception <= 1;
        _eret <= 0;
        cause <= 5'b01101;
end
end
    _break:
begin
    exception <= 1;
    _eret <= 0;
    cause <= 5'b01001;
end
    eret:
begin
    exception <= 0;
    _eret <= 1;
end
    syscall:
begin
    exception <= 1;
    _eret <= 0;

```

```

        cause <= 5'b01000;
    end
    default:
    begin
        exception <= 0;
        _eret <= 0;
        cause <= 5'b00000;
    end
endcase
if(moreop == mfc0)
begin
    mfc0src <= 1;
    mtc0src <= 0;
end
else if(moreop == mtc0)
begin
    mfc0src <= 0;
    mtc0src <= 1;
end
else
begin
    mfc0src <= 0;
    mtc0src <= 0;
end
end
endmodule

```

2.5regfile

```

`timescale 1ns / 1ps
module regfile(
    input clk, rst,
    input [4:0] raddr1,
    input [4:0] raddr2, //所需读取的寄存器的地址
    input [4:0] waddr,
    input [31:0] wdata,
    input regfilesrc,//写寄存器的地址//写寄存器数据, 数据在 clk 下降沿时被写入
    output [31:0] rdata1,
    output [31:0] rdata2,//raddr1 所对应寄存器的输出数据 //raddr2 所对应寄存器的输出
    data
    output [31:0] ans
);
reg [31:0] array_reg [31:0]; //寄存器

```

```

//写寄存器
integer i;
always @(posedge clk)
begin
    if(rst)
        begin
            for(i=0;i<32;i=i+1)
                array_reg[i] <= 0;
        end
    else
        begin
            if((waddr!=0)&&regfilesrc==1)
                begin
                    array_reg[waddr] <= wdata;
                end
            else
                begin
                    array_reg[waddr] <= array_reg[waddr];
                end
        end
    end
end

assign rdata1 = array_reg[raddr1];
assign rdata2 = array_reg[raddr2];
assign ans = array_reg[28];
endmodule

```

2.6cp0

```

module cp0(
    input clk,
    input rst,
    input mfc0,
    input mtc0,
    input [31:0]pc,
    input [4:0]Rd,//mfc0 读地址
    input [31:0]wdata,//mtc0 写数据
    input exception,//异常发生
    input eret,
    input [4:0]cause,//传递什么数据
    output [31:0]rdata,//mfc0 读数据
    output [31:0]exc_addr//返回给 pc 的地址
);

```

```

);

integer i;
reg[31:0]cp0[31:0];
reg[31:0] status_temp;
assign rdata = mfc0 ? cp0[Rd] : 32'hz;
assign exc_addr = (eret==1) ? cp0[14]:32'h00400004;
always@(negedge clk)
begin
    if(rst)
        begin
            for(i=0;i<32;i=i+1)
                cp0[i]<=0;
        end
    else
        begin
            if(mtc0)
                cp0[Rd] <= wdata;
            else
                if(exception)
                    begin
                        status_temp<=cp0[12];
                        if(eret==1'b0)
                            begin
                                cp0[12]<=cp0[12]<<5;
                                cp0[13]<={25'b0,cause,2'b0};
                                cp0[14]<=pc;
                            end
                        else
                            cp0[12]<=status_temp;
                    end
            end
        end
    end
endmodule

```

2.7hi_lo

```

`timescale 1ns / 1ps
module hi_lo(
    input clk,
    input rst,
    input [31:0] Write_hi,
    input [31:0] Write_lo,
    output [31:0] Rd_hi,

```

```

        output [31:0] Rd_lo
    );
    reg [31:0] hi;
    reg [31:0] lo;
    assign Rd_hi = hi;
    assign Rd_lo = lo;

    always @(posedge clk)
    begin
        if(rst)
        begin
            hi <= 32'b0;
            lo <= 32'b0;
        end
        else
        begin
            hi <= Write_hi;
            lo <= Write_lo;
        end
    end
endmodule

```

2.8branchpredict

```

`timescale 1ns / 1ps
module branchpredict(
    input clk,
    input [31:0] pc,
    input [31:0] inst,
    input stall,
    input [31:0] rs,
    input [31:0] alu_a,
    input [31:0] alu_b,
    output reg isbranch,
    output reg [31:0]branch_pc
);
    wire [16:0]moreop;
    assign moreop = {inst[31:21],inst[5:0]};
    wire [11:0]opcode;
    assign opcode = {inst[31:26],inst[5:0]};
    wire [5:0]halfop;
    assign halfop = {inst[31:26]};

```

```

parameter [5:0]
beq = 12'b000100,
bne = 6'b000101,
bgez = 6'b000001,

j = 6'b000010,
jal = 6'b000011;

parameter [11:0]
jr = 12'b000000_001000,
jalr = 12'b000000_001001;

always @(*)
begin;
  //isbracnh
  if(stall)
    begin
      isbranch <= 0;
      branch_pc <= 0;
    end
  else if(opcode==jr || opcode==jalr)
    begin
      isbranch <= 1;
      branch_pc <= rs-32'h00400000;
    end
  else if(halfop==j || halfop==jal)
    begin
      isbranch <= 1;
      branch_pc <= {pc[31:28],inst[25:0]<<2}-32'h00400000;
    end
  else if((halfop==beq && alu_a==alu_b) || (halfop==bne && alu_a!=alu_b) || (halfop==bgez && $signed(alu_a)>=0))
    begin
      isbranch <= 1;
      branch_pc <= pc+{{(32 - 18){inst[15]}},inst[15:0],2'b00}+32'b100;
    end
  else
    begin
      isbranch <= 0;
      branch_pc <= 0;
    end
end

```

```
endmodule
```

2.9 Execute

```
`timescale 1ns / 1ps
module Execute(
    input clk,
    input [31:0]exe_rs_in,
    input [31:0]exe_rt_in,
    input [4:0]exe_aluchoice,
    output [31:0]exe_alu_out,
    output [31:0]exe_hi_out,
    output [31:0]exe_lo_out
);
alu cpu_alu(clk, exe_rs_in, exe_rt_in, exe_aluchoice, exe_alu_out, exe_hi_out, exe_lo_out);

endmodule
```

2.10 alu

```
`timescale 1ns / 1ps
module alu(
    input clk,
    input [31:0] A,
    input [31:0] B,
    input [4:0] aluchoice,
    output reg [31:0] result,
    output reg [31:0] hi_result,
    output reg [31:0] lo_result
);
integer i; // CLZ 用
integer j;
integer max;
reg [63:0]tomux;
always @(*)
begin
    result <= 0;
    hi_result <= 0;
    lo_result <= 0;
    tomux <= 0;
    case(aluchoice)
        5'b00000: result <= A + B;                                //ADDU
        5'b00001: result <= $signed(A) + $signed(B);                //ADD
    endcase
end
```

```

5'b00010: result <= A - B;                                //SUBU
5'b00011: result <= $signed(A) - $signed(B);             //SUB
5'b00100: result <= A & B;                               //AND
5'b00101: result <= A | B;                               //OR
5'b00110: result <= A ^ B;                               //XOR
5'b00111: result <= ~(A|B);                            //NOR
5'b01000: result <= {B[15:0], 16'b0};                  //LUI
5'b01001: result <= (A < B) ? 1 : 0;                  //SLTU
5'b01010: result <= ($signed(A)<$signed(B)) ? 1 : 0; //SLT
5'b01011: result <= $signed(B) >>> A;                //SRA 向右算

```

术移位

```
5'b01100: result <= B >> A;                           //SRL 向右
```

逻辑移位

```
5'b01101: result <= B << A;                           //SLL SLR
```

```

5'b10001:
begin//DIV
    hi_result <= $signed(A) % $signed(B);
    lo_result <= $signed(A) / $signed(B);
end
5'b10010:
begin//DIVU
    hi_result <= A % B;
    lo_result <= A / B;
end
5'b10011:
begin// MUL
    tomux = $signed(A) * $signed(B);
    hi_result = tomux[63:32];
    lo_result = tomux[31:0];
end
5'b10100:
begin//MULTU
    tomux = A * B;
    hi_result = tomux[63:32];
    lo_result = tomux[31:0];
end
5'b10101://CLZ
begin
    j <= 0;
    max <= 0;
    for(i = 31; i >= 0;i = i-1)
begin
    if(A[i]==1'b1)

```

```

        j <= 1;
        if(!j)
            max <= max + 1;
        end
        result <= max;
    end
endcase
end
endmodule

```

2.11Memory

```

`timescale 1ns / 1ps
module Memory(
    input clk,
    input [1:0] mem_dmem_inchoice,
    input [31:0] mem_dmem_addr,//输入输出用一个地址
    input [31:0] mem_dmem_in,
    input [2:0] mem_dmem_outchoice,
    output [31:0] mem_dmem_out
);

    dmem cpu_dmem(
        clk,
        mem_dmem_inchoice,
        mem_dmem_addr,
        mem_dmem_in,
        mem_dmem_outchoice,
        mem_dmem_out
    );
endmodule

```

2.12dmem

```

`timescale 1ns / 1ps
module dmem(
    input clk,//时钟
    input [1:0]dmem_inchoice,
    input [31:0] addr,//地址
    input [31:0] data_in,//输入
    input [2:0]dmem_outchoice,
    output reg [31:0] data_out//输出
);

```

```

wire [7:0]Addr;
assign Addr = addr[7:0];
wire [7:0]lh_lb;
reg [7:0] num [0:1023];//寄存器
assign lh_lb = num[Addr];
always@(posedge clk)
begin
    case (dmem_inchoice)
        2'b00:num[Addr]<=num[Addr];
        2'b01:
        begin
            num[Addr]<=data_in[31:24];
            num[Addr+1]<=data_in[23:16];
            num[Addr+2]<=data_in[15:8];
            num[Addr+3]<=data_in[7:0];
        end
        2'b10:
        begin
            num[Addr]<=data_in[15:8];
            num[Addr+1]<=data_in[7:0];
        end
        2'b11:num[Addr]<=data_in[7:0];
    endcase
    case(dmem_outchoice)
        3'b000:
        begin
            data_out <= {num[Addr],num[Addr+1],num[Addr+2],num[Addr+3]};
        end
        3'b001:
        begin
            data_out <= {{16{lh_lb[7]}},num[Addr],num[Addr+1]};
        end
        3'b010:
        begin
            data_out <= {{16{1'b0}},num[Addr],num[Addr+1]};
        end
        3'b011:
        begin
            data_out <= {{24{lh_lb[7]}},num[Addr]};
        end
        3'b100:
        begin
            data_out <= {{24{1'b0}},num[Addr]};
        end
    end

```

```
      endcase  
    end  
endmodule
```

2.13Writeback

```

`timescale 1ns / 1ps
module Writeback(
    input [1:0]wb_rfaddr_inchoice,
    input [4:0]wb_rfaddr1,
    input [4:0]wb_rfaddr2,
    input [4:0]wb_rfaddr3,
    input [2:0]wb_rfinchoice,
    input [31:0]wb_rf1,
    input [31:0]wb_rf2,
    input [31:0]wb_rf3,
    input [31:0]wb_rf4,
    input [31:0]wb_rf5,
    input [31:0]wb_rf6,
    input wb_rf_inallow,
    input wb_hi_inchoice,
    input [31:0]wb_rs1,
    input [31:0]wb_alu_hiout,
    input wb_lo_inchoice,
    input [31:0]wb_rs2,
    input [31:0]wb_alu_loout,
    output [4:0]wb_rfaddr_out,
    output [31:0]wb_rf_out,
    output wb_rf_allow,
    output [31:0]wb_hi_out,
    output [31:0]wb_lo_out
);
assign wb_rfaddr_out = wb_rfaddr_inchoice==2'b00 ? wb_rfaddr1 :
    (wb_rfaddr_inchoice==2'b01 ? wb_rfaddr2 : wb_rfaddr3);
assign wb_rf_out = wb_rfinchoice==3'b000 ? wb_rf1 :
    (wb_rfinchoice==3'b010 ? wb_rf2 :
    (wb_rfinchoice==3'b011 ? wb_rf3 :
    (wb_rfinchoice==3'b100 ? wb_rf4 :
    (wb_rfinchoice==3'b110 ? wb_rf5 :
    (wb_rfinchoice==3'b111 ? wb_rf6 : 0)));

```

```

(wb_rfchoice==3'b101 ? wb_rf5 : wb_rf6))));

assign wb_rf_allow = wb_rf_inallow;
assign wb_hi_out = wb_hi_inchoice ? wb_rs1 : wb_alu_hiout;
assign wb_lo_out = wb_lo_inchoice ? wb_rs2 : wb_alu_loout;
endmodule

```

2.14seg7x16

```

`timescale 1ns / 1ps
module seg7x16(
    input clk,
    input reset,
    input [31:0] i_data,
    output [7:0] o_seg,
    output [7:0] o_sel
);

parameter period= 10000;
reg [31:0]cnt;
reg clk_out;
always @(posedge clk,posedge reset)      //分频 50Hz
begin
if (reset)
    cnt <= 0 ;
else begin
    cnt<= cnt+1;
    if (cnt== (period >> 1) - 1)
        clk_out <= #1 1'b1;
    else if (cnt == period - 1)
        begin
            clk_out <= #1 1'b0;
            cnt <= #1 'b0;
        end
    end
end
end

wire seg7_clk = clk_out;

reg [2:0] seg7_addr;

always @ (posedge seg7_clk, posedge reset)
if(reset)

```

```
    seg7_addr <= 0;  
  else  
    seg7_addr <= seg7_addr + 1'b1;
```

```
reg [7:0] o_sel_r;
```

```
always @ (*)  
  case(seg7_addr)  
    7 : o_sel_r = 8'b01111111;  
    6 : o_sel_r = 8'b10111111;  
    5 : o_sel_r = 8'b11011111;  
    4 : o_sel_r = 8'b11101111;  
    3 : o_sel_r = 8'b11110111;  
    2 : o_sel_r = 8'b11111011;  
    1 : o_sel_r = 8'b11111101;  
    0 : o_sel_r = 8'b11111110;  
  endcase
```

```
reg [31:0] i_data_store;  
always @ (posedge clk, posedge reset)  
  if(reset)  
    i_data_store <= 0;  
  else  
    i_data_store <= i_data;
```

```
reg [7:0] seg_data_r;  
always @ (*)  
  case(seg7_addr)  
    0 : seg_data_r = i_data_store[3:0];  
    1 : seg_data_r = i_data_store[7:4];  
    2 : seg_data_r = i_data_store[11:8];  
    3 : seg_data_r = i_data_store[15:12];  
    4 : seg_data_r = i_data_store[19:16];  
    5 : seg_data_r = i_data_store[23:20];  
    6 : seg_data_r = i_data_store[27:24];  
    7 : seg_data_r = i_data_store[31:28];  
  endcase
```

```
reg [7:0] o_seg_r;  
always @ (posedge clk, posedge reset)  
  if(reset)  
    o_seg_r <= 8'hff;  
  else  
    case(seg_data_r)
```

```
4'h0 : o_seg_r <= 8'hC0;  
4'h1 : o_seg_r <= 8'hF9;  
4'h2 : o_seg_r <= 8'hA4;  
4'h3 : o_seg_r <= 8'hB0;  
4'h4 : o_seg_r <= 8'h99;  
4'h5 : o_seg_r <= 8'h92;  
4'h6 : o_seg_r <= 8'h82;  
4'h7 : o_seg_r <= 8'hF8;  
4'h8 : o_seg_r <= 8'h80;  
4'h9 : o_seg_r <= 8'h90;  
4'hA : o_seg_r <= 8'h88;  
4'hB : o_seg_r <= 8'h83;  
4'hC : o_seg_r <= 8'hC6;  
4'hD : o_seg_r <= 8'hA1;  
4'hE : o_seg_r <= 8'h86;  
4'hF : o_seg_r <= 8'h8E;  
endcase  
  
assign o_sel = o_sel_r;  
assign o_seg = o_seg_r;  
  
endmodule
```